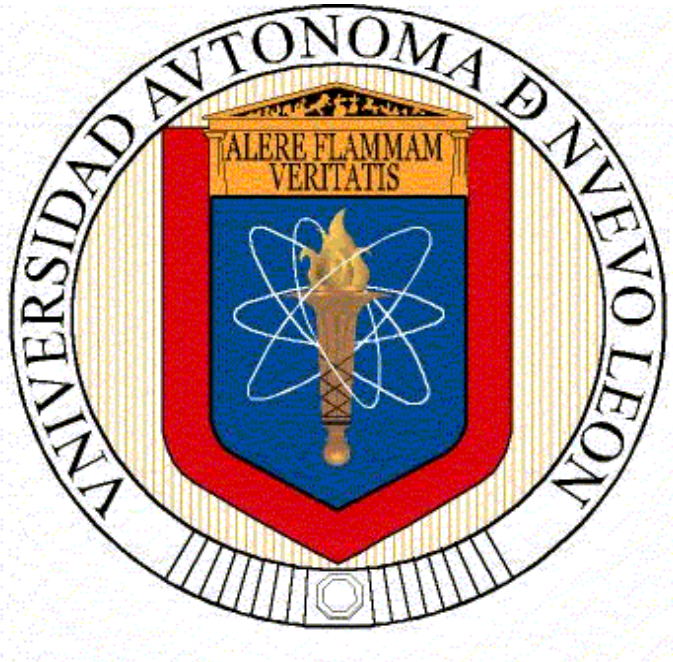
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS



MODELACIÓN DE REDES DE REGULACIÓN GENÉTICA CON REDES NEURONALES RECURRENTES DESDE UNA PERSPECTIVA BAYESIANA

POR
EDGAR JIMÉNEZ PEÑA

COMO REQUISITO PARA OBTENER EL GRADO DOCTOR EN CIENCIAS CON ORIENTACIÓN EN MATEMÁTICAS

MAYO, 2016

Modelación de redes de regulación genética con redes neuronales recurrentes desde una perspectiva bayesiana

Los miembros del comité aprueban la tesis de doctorado de Edgar Jiménez Peña.

Dra. María Aracelia Alcorta García _____

Dr. Francisco Javier Almaguer Martínez _____

Dr. José Arturo Berrones Santos _____

Dr. Omar Gónzalez Amezcua _____

Dr. Jorge Luis Menchaca Arredondo _____

Dedicada a:

Dione Rivera, mi esposa, quien sin ella esta tesis sería solo un sueño, y a mi madre, quién me dio las alas para este camino.

Agradecimientos

A mis asesores: Dra. María Aracelia Alcorta, Dr. Arturo Berrones y Dr. Javier Almaguer cuyo apoyo y orientación fueron invaluables a lo largo del doctorado y durante la elaboración de mi tesis.

A los miembros del Comité de Tesis: Dra. María Aracelia Alcorta, Dr. Arturo Berrones, Dr. Javier Almaguer, Dr. Omar Gónzalez y Dr. Jorge Luis Menchaca por su ayuda en el mejoramiento de la calidad de la tesis.

Resumen

Modelación de redes de regulación genética con redes neuronales recurrentes desde una perspectiva bayesiana

Publicación No. _____

Edgar Jiménez Peña

Universidad Autónoma de Nuevo León

Facultad de Ciencias Físico Matemáticas

Profesores co-asesores: Dra. María Aracelia Alcorta García,

Dr. Francisco Javier Almaguer Martínez, Dr. Arturo Berrones

DCOM, 2015

En el siguiente trabajo se desarrolló una metodología para la creación de modelos generales dinámicos de redes de regulación genética (GRN). El entendimiento de estas redes permitirá aumentar el conocimiento de procesos celulares fundamentales tales como ciclo celular, diferenciación celular, apoptosis, etc. En los últimos años ha aumentado la disponibilidad de series de tiempo de datos de expresión genética, lo cual permite el estudio de genomas y no solo de pares de genes o un conjunto reducido de ellos. Este conocimiento es de vital importancia para genetistas y biólogos debido a su rol en el metabolismo de un organismo. Sin embargo el comportamiento de estos datos presenta efectos no lineales que hacen difícil su estudio y por tanto las inferencias de las relaciones entre genes. La metodología desarrollada se basa en redes neuronales recurrentes en tiempo continuo, las cuales dependen de una suavización de los datos de las series de tiempo a modelar. Esta suavización permite estimar dinámicas de campo medio de las variables, por lo que la solución de las ecuaciones diferenciales se sustituye por un suavizado bayesiano, lo que lleva a un ahorro en el tiempo de cómputo para la estimación de los parámetros de la red

. El método se probó en datos reales de regulación genética y tiene un mejor desempeño que otros formalismos publicados, medido a través del error cuadrado medio de los datos de expresión .

Índice general

Capítulo 1

Introducción

1.1. Introducción

La inferencia de redes de regulación genética es un tema con un fuerte proceso de investigación en muchas universidades y centros de investigación, esto en parte debido a que la cantidad de información disponible se obtiene de manera más sencilla y en mayor cantidad de organismos. Estos datos se refieren a diferentes niveles del proceso celular: concentración de proteínas, concentración de mRNA, etc. Sin embargo establecer las relaciones entres los diferentes genes aún se encuentra en fase de estudio, pues el planteamiento de modelos, su resolución y los experimentos derivados de estas conclusiones han dado resultados con poco nivel de precisión en la mayoría de los casos. Por este tipo de resultados se siguen planteando diferentes modelos, así como algoritmos de cálculo para éstos. Por ello la aportación de esta tesis se centrará en el planteamiento de un modelo que tenga las siguientes características: adecuado para la representación de estas redes, con la capacidad de obtener inferencias acerca de la relación entre los diferentes genes y que además sea aplicable a diferentes fuentes de información.

1.2. Antecedentes

Las redes de regulación genética son un tema de relevancia actual para la biología, por lo que su estudio se ha llevado a cabo desde diferentes vertientes [2, 5, 12]. Una de estas forma de estudio corresponde al estudio de proteínas específicas asociadas a un gen con mediciones realizadas en diferentes puntos del tiempo.

Dentro de esta rama se han realizado diferentes aproximaciones para el planteamiento y resolución de un modelo que permita hacer una estimación de las diferentes interacciones entre los genes que conforman una red de regulación genética. Varias de estas aproximaciones se mencionan en [12, 40], donde diferentes tipos de modelos y algoritmos de solución se aplican a diferentes conjuntos de datos para encontrar las interacciones entre estos genes.

Uno de los trabajos que plantea un modelo lo suficientemente general para modelar diferentes tipos de interacciones corresponde al trabajo de [25] que utiliza como modelo una red neuronal recurrente, la cual resuelve mediante una optimización basada en partículas. Otro de los trabajos que hace mención a uno de los temas presentados en esta tesis corresponde a [20] que utiliza un esquema de ecuaciones diferenciales y un proceso de suavización previo durante el proceso de optimización. Esta técnica de suavizado es similar a la usada en esta tesis, pero los principios de justificación son muy diferentes, además de que se usa de manera conjunta durante el proceso de optimización, mientras que en el presente trabajo se utiliza de manera separada la suavización y la optimización que hacen más sencillo el problema de optimización.

1.3. Motivación

Las redes de regulación genética son un tema de interés actual puesto que su entendimiento llevará a una mejor comprensión de las rutas metabólicas para el combate de diferentes enfermedades, sin embargo estas corresponden a un proceso complejo donde existen múltiples interacciones a diferentes niveles entre los genes, proteínas y mRNA mensajero entre otros. Además de que es un proceso dinámico donde las concentraciones de cada uno de estos elementos es cambiante. Por otra parte existe la limitante de que la medición de estos elementos es compleja y requiere de material especializado, por lo que las muestras temporales del proceso son pequeñas y están sujetas a ruidos de medición. De ahí el interés de desarrollar un método que permita hacer inferencia acerca de las conexiones de redes de regulación genética a través del estudio de muestras pequeñas temporales.

1.4. Aportaciones

A continuación se presentan las aportaciones de la tesis.

El desarrollo y calibración de un procedimiento para la inferencia de redes de regulación genética mediante el uso de redes neuronales en tiempo continuo el cual se publicó en [52], además de presentarse como una exposición en el CLAIO XVII en Monterrey Nuevo León.

Entre las aportaciones específicas derivadas de este trabajo se encuentran: la propuesta de una descripción de campo medio para el nivel de expresión genética en un tiempo fijo, lo cual permite realizar un suavizado de los datos para su representación posterior; otra es el suavizado basado en transformadas de Fourier, donde la estimación de los parámetros de la serie de Fourier se estiman mediante un procedimiento de minimización de función del error derivado de un planteamiento bayesiano, además se comparó el método desarrollado en este trabajo con otros métodos de suavizado para muestras pequeñas y con ruido de medición, donde se obtuvo que el método planteado tuvo un mejor desempeño que sus alternativas de comparación.

Una aportación clave en el trabajo es el uso del suavizado de datos para el entrenamiento de la red neuronal recurrente, este procedimiento, como se mencionó, es derivado de un planteamiento bayesiano de las observaciones, esto permitió mejorar el tiempo de procesamiento necesario para el entrenamiento de la red neuronal debido a la disminución de evaluaciones de la función objetivo. Esta aportación se ve sustentada mediante un experimento de comparación contra otro método competitivo donde se controlan diferentes factores que afectan el tiempo de procesamiento. Por otra parte el entrenamiento de la red se basa en un planteamiento de una función de error de los gradientes usando los datos calculados a partir de la red neuronal recurrente y de los datos suavizados.

Por último se presenta como aportación la comparación de la metodología propuesta, redes neuronales recurrentes más suavización, contra otros procedimientos sobre los mis-

mos conjuntos de datos obteniendo una mejora en los niveles de error para datos dentro y fuera de muestra.

En resumen la metodología propuesta con respecto a alternativas competitivas requiere una menor cantidad de evaluaciones de la función objetivo, lo que se traduce en un disminución del tiempo de procesamiento necesario para la estimación del modelo. Por otra parte se encuentra que el error de ajuste de este método para datos fuera de muestra es menor que para otros métodos, para diferentes conjuntos de datos de redes de regulación genética, lo que lo coloca como una alternativa viable para el estudio de este tipo de redes.

1.5. Organización de la Tesis

Esta tesis se organiza de la siguiente manera. En el capítulo 2 se hace una revisión de los conceptos de redes de regulación genética, los tipos de modelos que se han usado para estudiarlas, así como los algoritmos que se han empleado para solucionar dichos modelos, así como una revisión de las diferentes fuentes de información disponibles. Posteriormente se explicará el uso de las redes neuronales en tiempo continuo en el área de redes de regulación genética.

En el capítulo 3 se menciona el uso de las redes neuronales en tiempo continuo en trabajos anteriores, especialmente el trabajo de [25] donde se realizó una aplicación de este modelo mediante un método de solución basado en PSO (optimización basada en enjambres de partículas por sus siglas en inglés). Posteriormente se describe la metodología propuesta que tiene como base una red neuronal recurrente, cuya estimación de parámetros se realiza mediante un método basado en la suavización de los datos muestrales. Esta técnica esta basada en la suposición de que la expresión genética en un tiempo fijo puede representarse como una expresión de campo medio, con lo que las derivadas correspondientes de la concentración del sistema pueden aproximarse mediante una adecuada suavización de los datos. Para encontrar el mejor método de suavización se realiza una serie de experimentos sobre datos simulados para encontrar el mejor método de suavización cuando se tienen pocos datos. Con esta información se aplica el método a estos datos suavizados y se reportan los resultados. Posteriormente para realizar una comparación con una alternativa razonable de resolución para este tipo de problemas, se planteó un experimento de comparación con un algoritmo evolutivo, donde las medidas a contrastar fueron velocidad y error obtenido. Al final del capítulo se muestra una comparación del método propuesto contra otros métodos de la literatura usando diferentes conjuntos de datos reportados en la literatura [20] y [42, 43].

En el capítulo 4 se comentan los resultados y se hace un serie de recomendaciones para trabajo futuro en esta línea. Además en los apéndices se muestra la publicación realizada con el trabajo de la presente tesis, así como las pruebas estadísticas resultado de la comparación del método propuesto contra el algoritmo evolutivo.

Capítulo 2

Marco Teórico

2.1. Redes de regulación genética

2.1.1. Definición de red de regulación genética

Las células realizan diferentes procesos en muchas y diferentes situaciones por medio de un conjuntos de genes, que definiremos de manera amplia como redes de genes interactuantes, proteínas y metabolitos [1]. De manera formal, una red de regulación genética (RRN) es un conjunto de segmentos de ADN que interaccionan entre ellos (de manera indirecta a través de su ARN y productos de expresión proteínica) y con otras sustancias en la célula, para de esta manera controlar la velocidad a las cuales los genes en la red se transcriben en mARN. En general, cada molécula de mARN genera una proteína específica (o conjunto de proteínas). Estas proteínas generadas puedes ser estructurales, enzimáticas (que catalizan reacciones), sin embargo ciertas proteínas solo sirven para activar otros genes, y dichas proteínas se denominana factores de transcripción que son los agentes principales en redes de regulación. Estas proteínas al unirse a la región del promo-

tor que se encuentra al inicio de otros genes los activan, iniciando la producción de otra proteína y así sucesivamente, mientras que otros factores de transcripción son inhibitorios [2]. En la Figura 2.1 se muestra una representación del proceso de transcripción, donde las proteínas se encuentran fuera del núcleo en el citoplasma de la célula.

La importancia de estas redes se debe a que muchos procesos celulares tales como ciclo celular, diferenciación celular y apoptosis son controlados mediante la regulación de genes, lo cual es esencial para todos los virus, procariontes y eucariotes. Este proceso les permite transformar la información codificada en los genes en proteínas para incrementar la versatilidad y adaptabilidad de un organismo permitiendo que la célula exprese la proteína cuando sea necesario. [2]

2.1.2. Modelos biológicos

El modelo biológico que soporta a una red de regulación genética se observa en la Figura 2.2 [4], donde los diferentes elementos que la constituyen y los procesos que los conectan se encuentran representados.

2.1.3. Formas de medición de los elementos de regulación genética

La cuantificación de los elementos en un proceso de regulación biológica depende de la etapa en la que se encuentra y de los elementos que se van a medir. La Tabla 2.1 muestra las tecnologías y elementos de medición para en una red de regulación genética [5] .

Cada una de las diferentes técnicas de medición se encuentra asociada a una etapa específica del proceso de regulación, así como a un actor específico. Las etapas y técnicas más usadas en las redes de regulación genética corresponden a elementos de ARN, debido

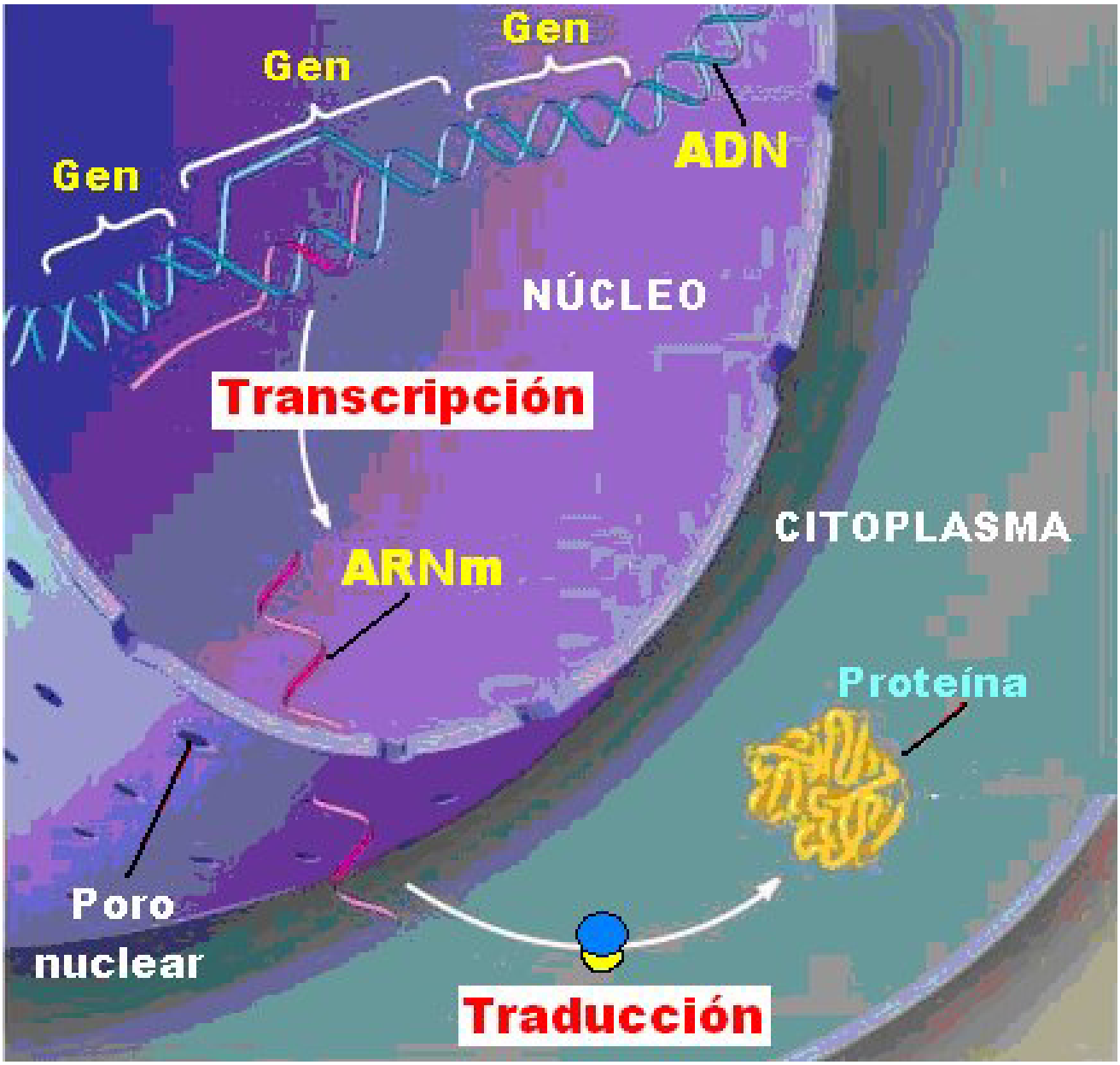


Figura 2.1: Proceso de transcripción de proteínas [3]

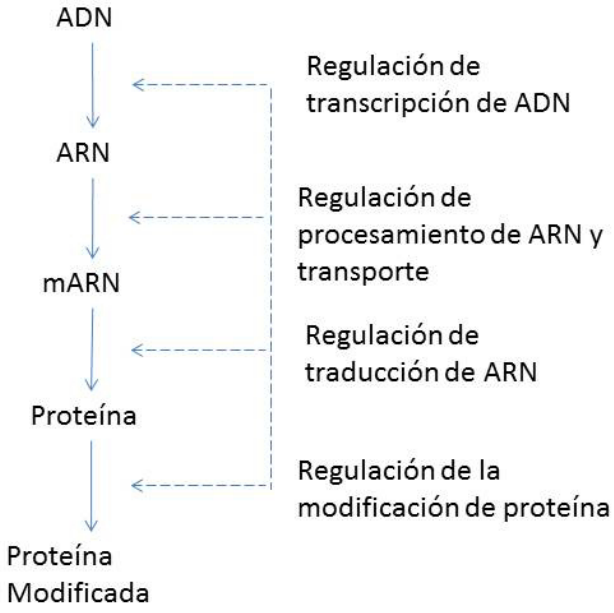


Figura 2.2: Regulación de la expresión genética en diferentes etapas de la síntesis de proteínas

Tabla 2.1: Elementos de medición de una red de regulación genética

Etapa	Objetivo	Proceso Regulatorio	Tecnología de medición
Dominios de Cromatina	ADN	Metilación de ADN	Arreglo/secuenciación de metilación, secuenciación ChIP, LC-MS
		Fosfolorización de ADN	
		Deacelitación de histonas	
Transcripción	ADN-ARN	Factor de transcripción	chip ChIP, secuenciación ChIP
		Represores	
		Activadores	
		Incrementadores	
Post-transcripción	ARN	Modificación de la caperuza	Microarreglos, secuenciación de ARN arreglo de exones, HITS-CLIP, secuenciación RIP
		Eliminación de intrones y empalme de exones	
		Poliadenilación	
		Edición de ARN	
		Silenciamiento de microARN	
Traducción	ARN-proteína	Iniciación de traducción	
		Elongación de péptidos	
		Terminación	
Post-Traducción	Proteína	Acilación	Arreglo de proteínas, LC-MS
		Fosforilización	
		Degradación de Proteínas	

a la dificultad que existe para medir proteínas específicas.

Entre las técnicas más populares para la medición de ARN se encuentran la técnica de los microarreglos, dicha técnica consiste en tomar diferentes muestras de tejidos para después utilizar un colorante fosforescente y posteriormente hibridizarlo en el microarreglo, el cual consiste en un conjunto de minisensores de ADN dispuestos en una plataforma sólida en forma de matriz. Posteriormente se hace un lavado y esto intensifica la señal fosforescente de los minisensores (productos PCR o iniciadores cortos de ADN) lo cual permite medirlos por medio de un scaner láser, con lo que se provee una medida semicuantitativa de la cantidad de moléculas de ácido nucleico que son complementarias a los sensores del microarreglo [6]. Una de las técnicas más usadas para la experimentación con microarreglos es el uso de muestras control y experimental, las cuales se hibridan en un mismo microarreglo para realizar su comparación, en la Figura 2.3 se muestran los pasos de este procedimiento [7].

2.1.4. Fuentes de información de redes de regulación genética

2.1.4.1. Conjuntos de datos reales de bases de información genética

Una de las metodologías más recientes para la medición de datos de interacción de proteínas-DNA corresponde a la inmonuprecipitación de cromatina (ChIP en inglés), esta metodología permite encontrar relaciones particularmente en los factores de transcripción y los promotores objetivos. Este procedimiento se centra en los factores de transcripción de epítopos conocidos que se encuentran unidos a fragmentos de ADN que contienen los promotores objetivos, a estos factores se les realiza la inmonuprecipitación de cromatina, para después hibridizar los fragmentos de DNA en un microarreglo intergenético.

Actualmente muchos datos de ChIP sobre levadura y otros organismos se encuentran

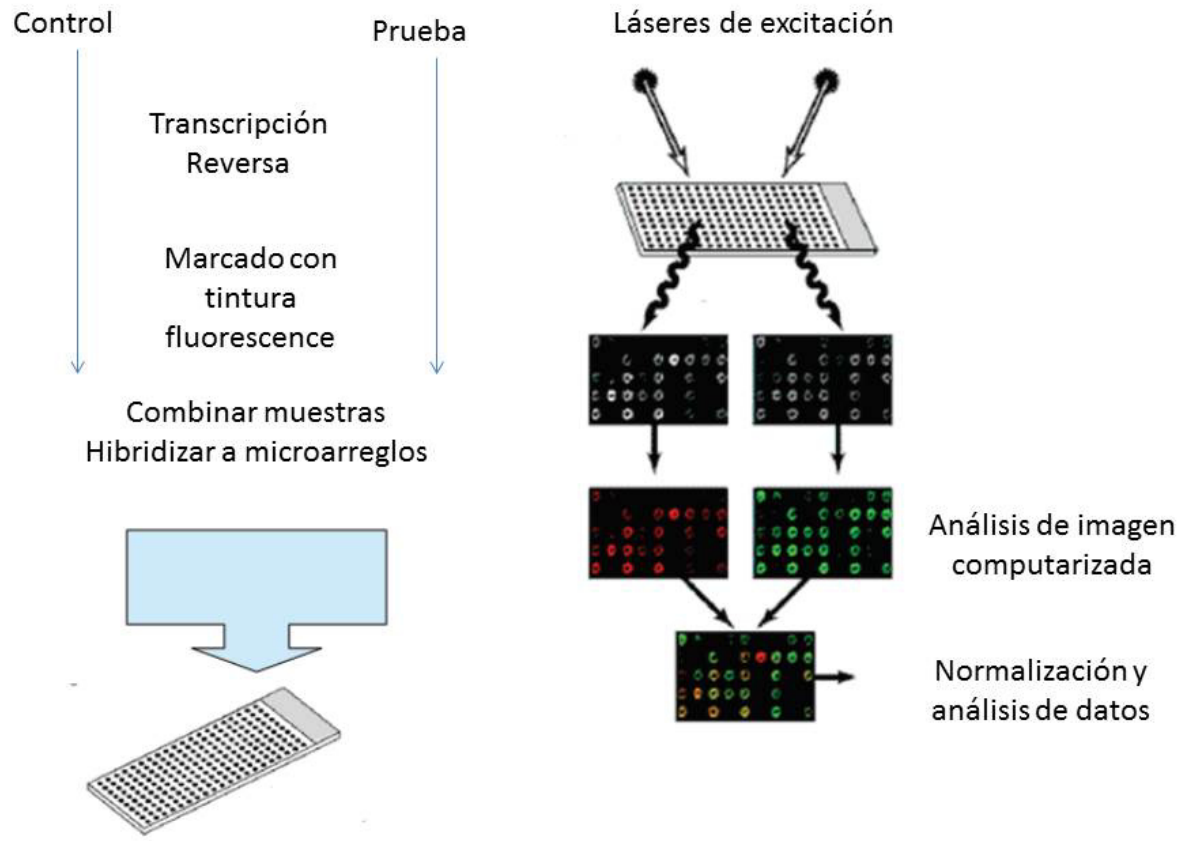


Figura 2.3: Diagrama de un experimento de microarreglos con dos muestras.

públicamente disponibles, por ejemplo los datos de la investigación de [8, 9] acerca de la localización de genes y se puede descargar de (http://web.wi.mit. edu/young/regulatory_network/).

Otro ejemplo es YEASTRACT (Búsqueda para reguladores transcripcionales y consenso sobre levadura) que es un repositorio de más de 12,5000 asociaciones regulatorias entre factores de transcripción y genes objetivo en la Saccharomyces cerevisiae. [2]

Uno de los conjuntos de datos más influyentes con respecto a la Saccharomyces cerevisiae fue el trabajo de [10] donde por medio de la técnica de microarreglos y muestras tomadas a diferentes tiempos, se realizó un análisis de más de 800 genes, esta información se puede consultar en http://genome-www.stanford.edu/cellcycle/. En [40] se muestra una comparación de diferentes métodos para recrear los datos de expresión del ciclo celular de la Saccharomyces Cerevisiae del conjunto de datos de Spellman. En él existen tres diferentes conjuntos de datos con diferente número de genes (6, 7 y 24). Varios autores mencionan que el trabajo [40] es actualmente uno de los estudios comparativos más exhaustivos con respecto a técnicas de inteligencia artificial aplicadas a modelos no lineales de las redes de regulación genética[26], [41]. Además [37] considera a [40] como un resumen detallado de algoritmos de reconstrucción aplicados a diferentes conjuntos de datos.

2.1.4.2. Bases sintéticas de información

Entre los esfuerzos que se han realizado para crear datos de comparación para la inferencia de redes de regulación genética, se encuentra la opción de crear redes artificiales que generen dinámicas semejantes a las de una red real. Entre estos esfuerzos se encuentra GeneNetWeaver [11] que genera datos de una red de regulación genética in silico a través de un modelo matemático subyacente y datos reales de redes de información genética. En la Figura 2.4 se puede observar la forma de este sistema. La generación final corresponde a series de tiempo que serán usadas para realizar la inferencia de la red que las generó.

Entre los puntos relevantes de este acercamiento se encuentra el uso de ecuaciones para la generación de las series de tiempo. Para cada gen i de la red, la tasa de cambio de concentración de mRNA F_i^{RNA} y la tasa de cambio de concentración de proteínas F_i^{Prot} se encuentran dados por:

$$\begin{aligned} F_i^{RNA}(\mathbf{x},\mathbf{y}) &= \tfrac{dx_i}{dt} &= m_i f_i(\mathbf{y}) - \lambda_i^{RNA} x_i \\ F_i^{Prot}(\mathbf{x},\mathbf{y}) &= \tfrac{dy_i}{dt} &= r_i x_i - \lambda_i^{Prot} y_i \end{aligned} \tag{2.1}$$

Donde m_i es la máxima tasa de transcripción, r_i es la tasa de traducción, λ_i^{RNA} y λ_i^{Prot} son las tasas de degradación de RNA y proteína respectivamente, y $\mathbf{x}, \mathbf{y}$ son los vectores que contienen todas las concentraciones de mRNA y proteínas respectivamente y $f_i(\cdot)$ es la función de activación de gen i que calcula la activación relativa del gen, que se encuentra entre 0 y 1. Parte de esta evidencia nos lleva a concluir que es necesario verificar los modelos matemáticos que se usan para la modelación de las redes complejas, sus supuestos y los resultados obtenidos de éstos.

2.2. Modelos matemáticos de representación de redes de regulación genética

Los modelos matemáticos de representación de las redes genéticas son variados y tienen diferentes representaciones. En [12] se describen los siguientes formalismos matemáticos para la representación de una red de regulación genética.

2.2.1. Redes Bayesianas

El primer formalismo discreto corresponde a las redes bayesianas, donde la estructura genética se representa mediante una gráfica acíclica dirigida $G = \langle V, E \rangle$. Los vértices

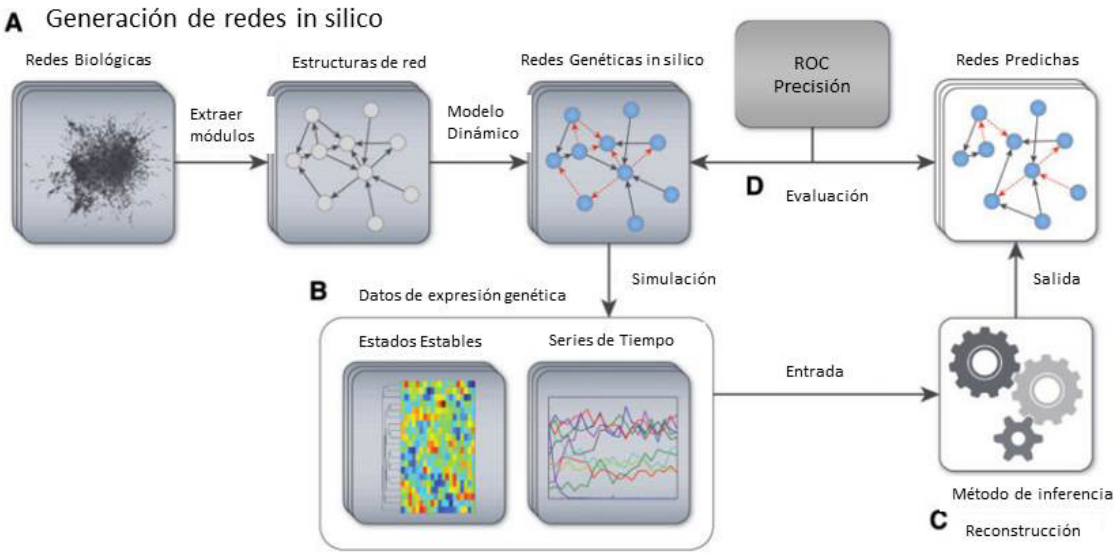


Figura 2.4: Representación del procedimiento NetWeaver para la generación de redes in silico y posterior benchmark.

$i \in V, 1 \leq i \leq n$ representan genes y tiene correspondencia con variables aleatorias X_i, donde esta variable representa el nivel de expresión del gen i. Para cada X_i, se define una distribución condicional $p(X_i \mid ancestros(X_i))$, donde $ancestros(X_i)$ denotan las variables correspondientes a los reguladores director de i en G. La gráfica G y las distribuciones condicionales $p(X_i \mid ancestros(X_i))$ definen a la red Bayesiana, la cual especifica de manera única una distribución de probabilidad conjunta $p(\mathbf{X})$. Esta gráfica contiene el supuesto de Markov, donde las distribuciones para un gen i en especifico son independientes de los $\sim ancestros(X_i)$, por lo que la distribución conjunta puede ser descompuesta en

$$p(\mathbf{X}) = \prod_{i=1}^{n} p(X_i \mid ancestros(X_i)) \tag{2.2}$$

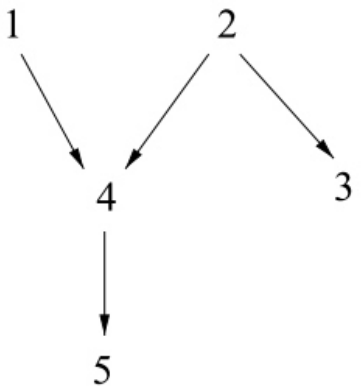
En la Figura 2.5 se muestra un ejemplo de una red de regulación genética

2.2.2. Ecuaciones diferenciales no lineales no ordinarias

En este formalismo se usa la concentración de mRNA, proteínas y otros elementos a través del uso de variables dependientes del tiempo. Las interacciones regulatorias toman la forma de relaciones funcionales y diferenciales entre las variables de concentración. De una manera más específica la regulación genética se modela por ecuaciones de velocidad de reacción que expresa la velocidad de producción de un producto genético (proteína, mRNA) como una función de las concentraciones de otros elementos del sistema. Estas ecuaciones tienen la forma

$$\frac{dx_i}{dt} = f_i(\mathbf{x}),\ x_i \geq 0,\ 1 \leq i \leq n \tag{2.3}$$

Esta ecuación puede ser extendida para utilizar concentraciones de elementos externos $u \geq 0$, por ejemplo, la alimentación de nutrientes externos, por lo que la ecuación anterior



$$p(X_1),\ p(X_2), p(X_4|X_1, X_2)$$

$$p(X_5|X_4), p(X_3|X_2)$$

$$p(\mathbf{X}) =$$
$$p(X_5|X_4)p(X_4|X_1, X_2)p(X_3|X_2)p(X_1)p(X_2)$$

$$i(X_1; X_2, X_3), i(X_2; X_1), i(X_4; X_3|X_1, X_2)$$

$$i(X_3; X_1, X_4, X_5|X_2), i(X_5; X_1, X_2, X_3|X_4)$$

Figura 2.5: Representación de una red de regulación genética a través de una Red Bayesiana, representada por una gráfica (imagen de la izquierda, donde los números representan nodos y las flechas las relaciones entre ellas). A la derecha se encuentran distribuciones de probabilidad condicional, la distribución de probabilidad conjunta y las independencias condicionales

se puede expresar como

$$\frac{dx_i}{dt} = f_i(\mathbf{x},\mathbf{u}),\ x_i \geq 0,\ 1 \leq i \leq n \tag{2.4}$$

También es posible incluir retrasos en el tiempo, los cuales surgen del tiempo necesario para completar los procesos necesarios para expresión de una proteína, por lo que la ecuación se puede expresar como

$$\frac{dx_i}{dt} = f_i(x_1(t-\tau_{i1}),...,x_n(t-\tau_{in})),\ x_i \geq 0,\ 1 \leq i \leq n \tag{2.5}$$

donde $\tau_{i1},...,\tau_{in} > 0$ representan retrasos temporales.

Algunas de las ecuaciones más usadas de este tipo hacen uso de términos de degradación y de excitación, las cuales tienen la forma

$$\begin{aligned} \tfrac{dx_1}{dt} &= \kappa_{1n} r(x_n) - \gamma_1 x_1,\ x_1 \geq 0 \\ \tfrac{dx_i}{dt} &= \kappa_{i(i-1)} x_{i-1} - \gamma_i x_i,\ x_i \geq 0,\ 1 \leq i \leq n \end{aligned} \tag{2.6}$$

donde los parámetros $\kappa_{1n}, \kappa_{21}, \kappa_{n.n-1} \geq 0$ corresponde a constantes de producción y $\gamma_1,...,\gamma_n \geq 0$ a constantes de degradación. En el caso de x_1 se encuentra una función de regulación no linear r. Un ejemplo de este tipo de función es la función de Hill que se expresa como

$$h^+(x_j,\theta_{ij},m) = \frac{x_j^m}{x_j^m + \theta_{ij}^m} \tag{2.7}$$

donde $\theta_{ij} > 0$ es el umbral de influencia de j en i y $m > 0$ es un parámetro de inclinación de la curva. La función toma valores entre 0 y 1 y se incrementa cuando $x_j \to \infty$, para $m > 1$ las curvas de Hill tienen una forma sigmoidal. En la Figura 2.6 se puede observar esta función junto con otras funciones de regulación.

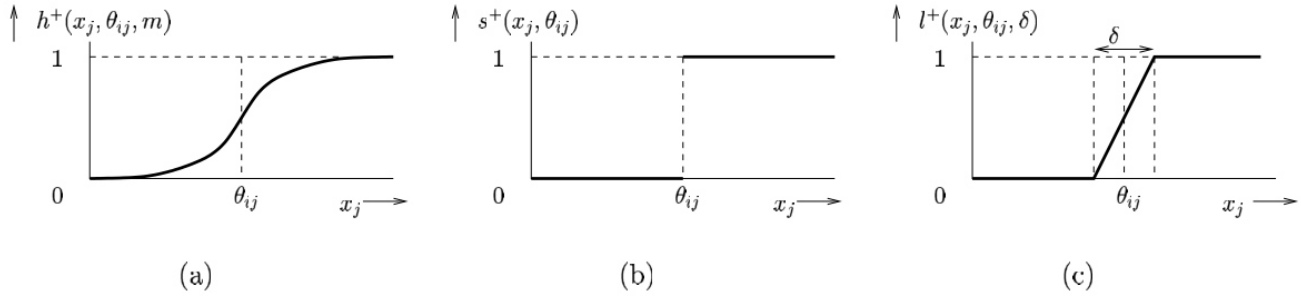


Figura 2.6: Ejemplos de funciones de regulación: (a) Función de Hill h^+, (b) Función de Heaviside o de escalón s^+, (c) Función logoide l^+

En [20] el modelo está definido por

$$\frac{\partial x_i}{\partial t}(t) = s_i - \gamma_i x_i(t) + \sum_{j=1}^{n} |\beta_{ij}| f_{ij}(x_j(t)) \tag{2.8}$$

donde $x_i(t)$ es la concentración del gen i en el tiempo t, s_i y γ_i son las tasas de síntesis y degradación basal para cada gen i. La variable β_{ij} denota la fuerza de la regulación del componente x_j sobre x_i y f_{ij} es la función de regulación correspondiente. $\beta_{ij} > 0$ es una activación, $\beta_{ij} < 0$ una inhibición, y $\beta_{ij} = 0$ significa que no hay regulación del gen j al gen i.

Un subcaso de las ecuaciones diferenciales no lineales corresponde al sistema S, que[21] considera un modelo base dentro de la literatura de procesos metabólicos pues se encuentran basados en la teoría de los sistemas bioquímicos.

Este sistema para representar las interacciones define la siguiente ecuación

$$\frac{dx_i}{dt} = \alpha_i \prod_{j=1}^{n} x_j^{g_{ij}} - \beta_i \prod_{j=1}^{n} x_j^{h_{ij}} \tag{2.9}$$

donde α_i es la la síntesis basal, β_i la tasa de degradación (constantes de cambio); g_{ij} indica la fuerza de la influencia del gen j en la síntesis del gen i y h_{ij} la influencia del gen j en

la degradación del gen i (órdenes cinéticos).

2.2.3. Ecuaciones estocásticas

El supuesto principal de las ecuaciones diferenciales es que las concentraciones de las sustancias varían continua y determinístcamente, lo cual es cuestionable en el caso de la regulación genética, por lo que ciertos autores han propuesto el uso de modelos discretos y estocásticos de regulación genética. En estos casos alternativos se considera que existe una cantidad discreta $\mathbf{X}$ de moléculas que se toman como variables de estado y que existe una probabilidad conjunta $p(\mathbf{X}, t)$, la cual expresa la probabilidad de que en el tiempo t se contenga X_1 moléculas de la primer sustancia, X_2 moléculas de la segunda sustancia y así sucesivamente. La evolución de la función $p(\mathbf{X}, t)$ se puede expresar como

$$p(\mathbf{X}, t + \triangle t) = p(\mathbf{X}, t)(1 - \sum_{j=1}^{m} \alpha_j \triangle t) + \sum_{j=1}^{m} \beta_j \triangle t \tag{2.10}$$

donde m corresponde al número de reacciones que pueden ocurrir en el sistema, $\alpha_j \triangle t$ es la probabilidad que la reacción j ocurrirá en el intervalo $[t + \triangle t]$ y $\beta_k \triangle t$ es la probabilidad que la reacción j llevará al sistema de un estado $\mathbf{X}$ a otro durante el intervalo $[t + \triangle t]$. Cambiando 2.10 y tomando el límite $\triangle t \to 0$ obtenemos la ecuación maestra

$$\frac{\partial}{\partial t} p(\mathbf{X}, t) = \sum_{j=1}^{m} (\beta_j - \alpha_j) p(\mathbf{X}, t) \tag{2.11}$$

Esta representación del sistema es más fácilmente entendible, sin embargo al involucrar las probabilidades es mucho más difícil de resolver por medios analíticos que una ecuación diferencial.

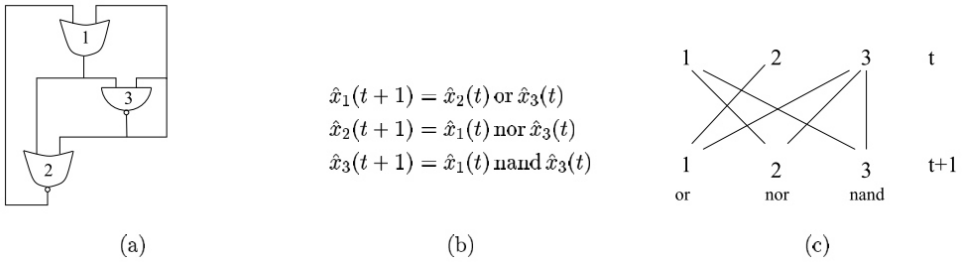


Figura 2.7: (a) Ejemplo de red booleana, (b) ecuaciones correspondiente. En este caso n=3, y k=2. (c) Diagrama de la red booleana

2.2.4. Redes booleanas

Este formalismo representa a la activación de un gen mediante una variable booleana (0 inactiva, 1 activa), por lo que la interacción entre los genes se puede representar mediante funciones booleanas que calculan el estado de activación de un gen de los estados de otros genes. En la Figura2.7 se puede observar un ejemplo de red booleana.

El vector $\hat{\mathbf{x}}$ de n-variables representa el estado de un sistema regulatorio de n variables, cada uno de los $\mathbf{x}_i$ tiene un valor de 1 o 0 por que el sistema tiene 2^n estados. El estado $\hat{x}_i$ de un elemento en el tiempo $t+1$ se calcula mediante una función booleana o regla $\hat{b}_i$ que se basa en el estado de k elementos del total de elementos en el tiempo t. En resumen, las dinámicas de un red booleana describiendo un sistema regulatorio está dado por (2.12).

$$\hat{x}_i(t+1) = \hat{b}_i(\mathbf{x}(t)), 1 \leq i \leq n \tag{2.12}$$

2.2.5. Métodos de teorías de información

Los métodos de teorías de información están basados en los conceptos de distancias y similitudes como lo describe [13]. El modelo más simple corresponde a una red simple

donde el peso de las conexiones corresponde a la correlación entre las variables. En este caso se considera que existe una interacción si el coeficiente de correlación es mayor a cierto nivel preestablecido. Otras medidas de distancias como medidas euclidianas, como información mutua, permite encontrar relaciones entre los diferentes genes.

2.2.6. Redes neuronales recurrentes en tiempo continuo

2.2.6.1. Conceptos Básicos

Las redes neuronales en tiempo continuo fueron introducidas por [14] de manera formal donde se comprobó que este formalismo aproxima la trayectoria finita de cualquier sistema dinámico con cualquier precisión deseada, y que cualquier curva continua puede ser aproximada por las salidas de una red neuronal.

El planteamiento básico de una red neuronal está determinado por la siguiente ecuación [14, 17]:

$$\frac{du_i(t)}{dt} = -\frac{u_i(t)}{\tau_i} + \sum_{j=1}^{m} w_{ij}\sigma(u_j(t)) + I_i(t), i = 1...m \tag{2.13}$$

donde $u_i(t)$ corresponde al estado de la unidad i, τ_i es la constante de tiempo de la unidad i, w_{ij} son los pesos de conexión entre las unidades y σ corresponde a la función de salida. Esta función de salida generalmente tiene una forma sigmoidal. Una representación de este sistema se muestra en la Figura 2.8.

2.2.6.2. Métodos de entrenamiento

Entre los métodos de solución que se encuentran está el recocido simulado [18], el BPTT y el PSO.

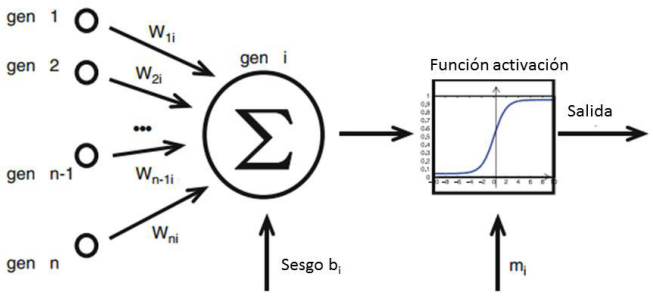


Figura 2.8: Representación gráfica de una red neuronal recurrente [17].

2.2.6.3. Usos en otras aplicaciones

Estas redes se han usado especialmente en las áreas de: robótica evolutiva, visión [15] y cooperación [16] especialmente.

2.3. Algoritmos de solución para los diferentes modelos matemáticos

En [40], que utiliza como base un modelo S, se mencionan diferentes métodos de optimización para calcular $\alpha_i, \beta_i, g_{ij}, h$: un algoritmo genético anidado con una estrategia evolutiva (GA + ES), un algoritmo iterativo basado en algoritmos genéticos (PEACE1), evolución diferencial como estrategia de búsqueda (DE + AIC), algoritmo genético para inferencia de parámetros (GA+ANN) y búsqueda local genética (GLDSC). Por otra parte en [20] el método de solución fue un modelo bayesiano cuyos datos se muestrearon con

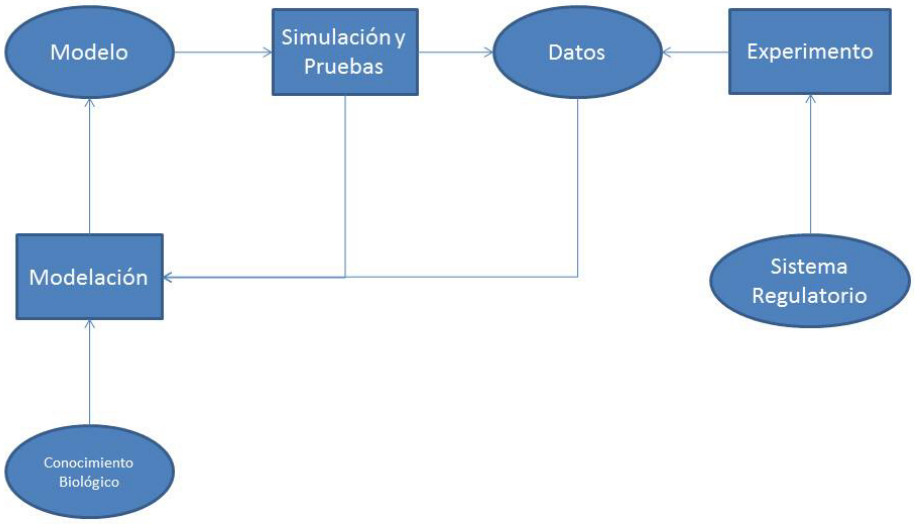


Figura 2.9: Proceso de análisis de regulación genética

una cadena de Markov Montecarlo (BSM).

2.4. Comparación de Análisis de sistemas de regulación genética

De acuerdo a [12] el análisis de regulación genética se puede resumir en la Figura 2.9. Este proceso de análisis se ha usado desde la década de 1960, pero con limitaciones debido a la cantidad de datos disponibles. En este momento la cantidad de información disponible permite realizar una mayor cantidad de modelaciones que en el pasado se encontraban restringidas por la cantidad de información.

La comparación de los resultados entre diferentes modelaciones, datos experimentales y técnicas de solución nos permite validar cuales son los enfoques más usuales de esta área en: conjuntos de datos, modelos matemáticos y técnicas de solución. Un resumen de estas combinaciones de datos, modelos y algoritmos se observa en la Tabla 2.2.

Tabla 2.2: Comparación de diferentes estudios por autor, tipo de datos, modelo usado y algoritmos de solución

Autor	Organismo	# de Genes	Genes seleccio-nados	Modelo	Algoritmo de Entrenamiento
D´Hasseler et al. (1999)	Rata	65	65	Ecuaciones diferenciales lineales	Mínimos cuadrados
Chen et al. (2001)	Levadura	6601	308	Métodos de teoría de información	Stepwise (recocido simulado)
Harteminke t al. (2002)	Levadura	6135	32	Red Bayesiana	Stepwise (recocido simulado)
Imoto et al. (2003)	Levadura	6000	36	Red Bayesiana	Stepwise (escalado)
Tamada et al. (2003)	Levadura	5871	124	Red Bayesiana	Máxima verosimilitud
Nariai et al. (2004)	Levadura	6178	99	Red Bayesiana	Stepwise (escalado)

Tabla 2.2: Comparación de diferentes estudios por autor, tipo de datos, modelo usado y algoritmos de solución

Autor	Organismo	# de Genes	Genes seleccionados	Modelo	Algoritmo de Entrenamiento
Basso et al. (2005)	Humano	10000	10000	Métodos de teoría de información	Fuerza bruta
Bernard y Hartermink (2005)	Levadura	6178	25	Red Bayesiana Dinámica	Stepwise (recocido simulado)
Guthke et al. (2005)	Humano	7619	6	Ecuaciones diferenciales lineales	Stepwise
Kimura et al. (2005)	T. Thermophilus	612	25	Modelo Sistema S	Algoritmo evolutivo
Bonneau et al. (2006)	Halobacterium	2400	531	Ecuaciones diferenciales lineales generalizada	Selección bivariada después LASSO
van Someren et al. (2006)	Ratón	9596	101	Ecuaciones diferenciales lineales	LASSO

Tabla 2.2: Comparación de diferentes estudios por autor, tipo de datos, modelo usado y algoritmos de solución

Autor	Organismo	# de Genes	Genes seleccio-nados	Modelo	Algoritmo de Entrenamiento
Faith et al. (2007)	E. Coli	4345	4345	Métodos de teoría de información	Fuerza bruta
Martin et al. (2007)	Ratón	34000	12	Red Booleana	Fuerza bruta
Koczan et al. (2008)	Humano	20	20	Ecuaciones diferenciales lineales	LASSO

Capítulo 3

Aplicación de redes neuronales en tiempo continuo a redes de regulación genética

3.1. Aplicaciones previas de redes neuronales en tiempo continuo(CTRNN)

En [18]se aplica una CTRNN sobre datos de levadura, de la Saccharommyces cerevisiae, y se logró ajustar los datos correspondientes a la curva seleccionada. Por otra parte [19] realiza una comparación entre diferentes metodologías de reconstrucción, entre ellas la CTRNN.

En [25] se realiza una aplicación de una CTRNN para inferir una red de regulación genética. El algoritmo para obtener los parámetros necesarios es un PSO (optimización a través de enjambre de partículas), este algoritmo se aplicó a datos sintéticos y reales,

con lo que se obtuvo una buena reconstrucción de la curvas de expresión. Este enfoque es semejante al usado en la sección 2.4, usando diferentes conjuntos de datos.

3.2. Descripción de la metodología

3.2.1. Bases teórica del planteamiento

El modelo inverso para el entendimiento de las interacciones de genes en un contexto dinámico puede plantearse de la siguiente manera: dada una serie de tiempo de expresión genética $\hat{x}_i(t)$, un modelo $x_i(t)$ puede ser inferido. Una función de error que depende del modelo y la muestra se define por,

$$E = \frac{1}{RTN} \sum_{r=1}^{R} \sum_{t=1}^{T} \sum_{i=1}^{N} [x_i(t) - \hat{x}_{i,r}(t)]^2. \tag{3.1}$$

Donde las sumas son sobre R muestras, T puntos temporales y N genes.

Un modelo para redes de regulación genética que se considera que captura de manera realista las interacciones de los genes dentro de la célula esta dada por una red neuronal recurrente en tiempo continuo [32, 33],

$$\tau_i \dot{x}_i = f\left[\sum_{j=1}^{N} w_{i,j} x_j + \sum_{k=1}^{K} v_{i,k} u_k + \beta_i\right] - \lambda_i x_i, \tag{3.2}$$

donde las cantidades w,v, β, τ y λ son parámetros (de ahora en adelante codificados por el vector $\boldsymbol{\theta}$), mientras que $\boldsymbol{x}$ y $\boldsymbol{u}$ representan niveles de expresión genética y variables externas, respectivamente.

La base de nuestro enfoque es proponer una descripción de campo medio para el nivel de expresión genética en un tiempo fijo

$$Q(\boldsymbol{x}, \boldsymbol{u}) = \prod_{i=1}^{N} q_i(x_i) \prod_{k=1}^{K} q_k(u_k), \tag{3.3}$$

con,

$$\int x_i q_i(x_i) dx_i = m_i \tag{3.4}$$

$$\int u_k q_k(u_k) du_i = \bar{u}_k \tag{3.5}$$

Los términos m_i y $\bar{u_i}$ dan una descripción de campo medio de la dinámica. Algunas funciones de verosimilitud adecuadas hacen uso de la suavización de las variables de campo medio, de tal manera que estas pueden ser definidas, por ejemplo, en términos del error

$$V(\boldsymbol{\theta}|\{\dot{\mathbf{m}}\}) = \frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} [< \dot{x}_i(t) >_Q - \dot{m}_i(t)]^2, \tag{3.6}$$

donde

$$\begin{aligned} \tau_i < \dot{x}_i >_Q = & \\ & f\left[\sum_{j=1}^{N} w_{i,j} m_j + \sum_{k=1}^{K} v_{i,k} \bar{u}_k + \beta_i\right] \\ & - \lambda_i m_i \end{aligned} \tag{3.7}$$

La densidad posterior de los parámetros de la red están dada por [34],

$$P(\boldsymbol{\theta}|\{\dot{\boldsymbol{m}}\}) = \frac{1}{Z} g(\boldsymbol{\theta}) h(V), \tag{3.8}$$

donde g es una densidad a priori y h una densidad de verosimilitud. Varios métodos pueden ser considerados para la obtención de información útil de los de los parámetros de la red a partir de (3.8). Por ejemplo, un modelo gaussiano con independientes a priori,

$$h(V) = \frac{1}{\sqrt{2\pi\sigma_\varepsilon^2}} \exp(-\frac{\varepsilon^2}{2\sigma_\varepsilon}), \tag{3.9}$$

$$P(\boldsymbol{\theta}|\{\dot{m}\}) = h(V)\prod_{r=1}^{R}\exp(-\frac{(\theta_r-\mu_r)^2}{2\pi\sigma_r})$$

con lo cual

$$-ln(P) = \frac{1}{2\sigma_\varepsilon}V + \frac{1}{2}\ln(2\pi\sigma_\varepsilon^2) + \sum_{r=1}^{R}\left\{\frac{1}{2\pi\sigma_r^2}(\theta_r-\mu_r)^2 + \frac{1}{2}\pi\sigma_r^2)\right\} \tag{3.10}$$

donde $\varepsilon = \sqrt{V}$. De las ecuaciones (3.9) y (3.10) los parámetros subyacentes pueden ser estimados de manera directa. El estimado MAP, por ejemplo, se obtiene directamente mediante la minimización directa de Eq.(3.10) con respecto a θ_r, μ_r y σ_r. De esta manera se deriva un modelo probabilistíco que asocia un valor medio y una varianza para los parámetros de interacción de la red. Esto se espera que sea más adecuado para la ingeniería inversa de la interacción de los genes que un estimador puntual.

3.2.2. Suavizamiento de los datos

La suavización de los datos requiere conocer cual es el mejor método de suavización, para comparar las diferentes metodologías de suavización se utilizó el siguiente esquema: generar datos de una red de regulación genética conocida (datos sintéticos), muestrear datos de la simulación, posteriormente suavizar la serie de datos y después entrenar la red neuronal recurrente en tiempo continuo mediante la función de error regularizada (3.1). Este procedimiento verificará el impacto del método de suavización y el método de muestreo mediante el usos de datos sintéticos de tal manera que se encuentre un tamaño de muestra recomendado así como un método de muestreo.

3.2.2.1. Datos sintéticos

Para generar los datos se utilizó el modelo de regulación genética Represilador [35] en su versión descrita en [36]. Las concentraciones de proteínas en el sistema están dadas por:

$$\begin{cases} i < 3 & \frac{dm_i}{dt} = -m_i + \frac{\alpha}{1+u_{i+1}^n} + \alpha_0 \\ i = 3 & \frac{dm_i}{dt} = -m_i + \frac{\alpha}{1+u_1^n} + \alpha_0 \end{cases} i = 1, 2, 3 \tag{3.11}$$

$$\frac{du_i}{dt} = -\beta(u_i - m_i)$$

donde las u_i son proporcionales a la concentración de las proteínas mientras que las m_i son proporcionales a las concentraciones de mRNA correspondientes a esas proteínas. El sistema muestra un comportamiento cíclico con un período de estabilización con respecto a las condiciones iniciales como se muestra en Figura 3.1.

3.2.2.2. Impacto del tamaño de la muestra y método de suavización

Para representar las condiciones experimentales se añadió un término estocástico en cada punto de datos que se distribuye de manera gaussiana con media = 0 y varianza σ_p^2 , este término es la varianza del proceso para cada proteína, este procedimiento se usa de manera común cuando se usan datos artificiales para aproximarlos a datos reales[37]. Para cuantificar el impacto del tamaño de muestra y método de suavización usamos el indicador $\chi^2_{red} = \frac{1}{n}\sum_{i=1}^{n} \frac{(O-E)^2}{\sigma_p^2}$ donde O es el valor verdadero de la serie, E es el valor suavizado, σ_p^2 es la varianza del proceso y n es el número de mediciones en la muestra. Se calculó este indicador en cada una de las proteínas u_i. El tamaño de la muestra es importante debido a que cada punto muestral tiene un costo asociado, el cual es generalmente alto para perfiles de expresión genética [38]. Los métodos de suavización que se compararon

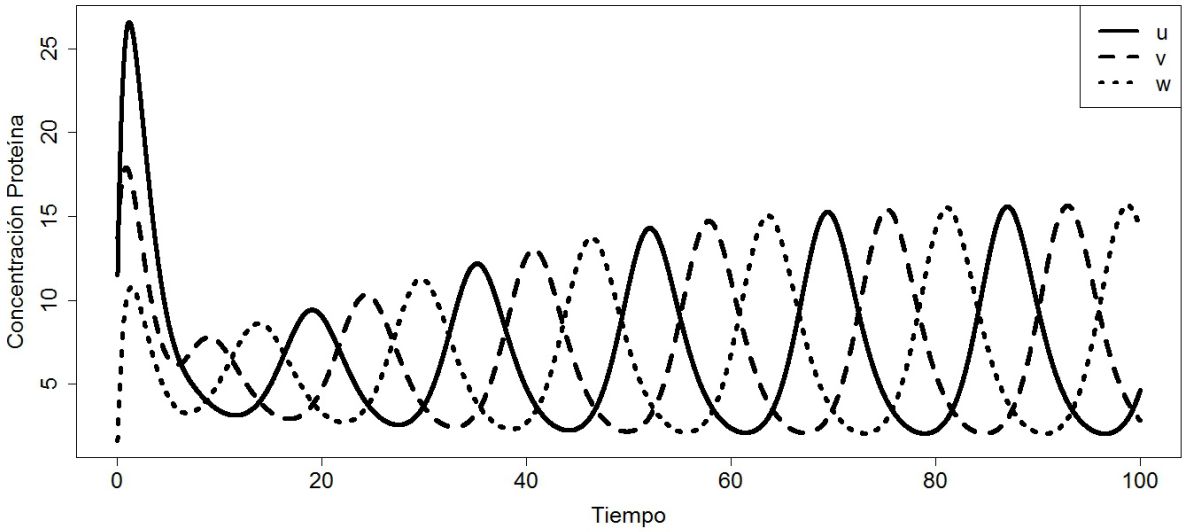


Figura 3.1: Concentraciones de proteínas u_i (u, v y w)

fueron: spline de suavización, LOESS, Kernel and suavización MAP basada en series de Fourier, la cual es una contribución original de este trabajo.

3.2.2.3. Suavización por MAP

Este método de suavización está basado en una serie de Fourier de la forma $f(t, \hat{a}, L) = \sum_{l=1}^{L} a_l \cos(l\frac{2\pi}{F}t)$ donde la estimación de los parámetros a_l es aproximadamente bayesiana. Se consideró que los parámetros a_l son probabilistícos de tal manera que

$$P(a_l) \to P(f_t) \Rightarrow < f >= \int f(t)P(f_t)df_t, \tag{3.12}$$

con una varianza asociada

$$\sigma_t^2 =< f^2 > - < f >^2 . \tag{3.13}$$

Suponiendo una muestra M y a una distribución a priori uniforme, se propone la función de probabilidad posterior:

$$P(\vec{a}_l|M) = \frac{1}{z}h(\mu|\vec{a}_l), \tag{3.14}$$

donde h es una densidad de verosimilitud y z es un factor de normalización. La función de densidad de verosimilitud es

$$h(\mu|\vec{a}_l) = \prod_{m=1}^{\mu} N[f(\hat{a}_l), \sigma_m^2] \tag{3.15}$$

que lleva a

$$\ln P = -\frac{1}{\sigma^2}\sum_{m=1}^{M}[\hat{y_m} - f(\hat{a}_l)]^2 - \frac{\mu}{2}(2\pi\sigma^2) \tag{3.16}$$

Minimizamos la expresión(3.16) mediante un método de gradiente conjugado. Con una revisión sistemática de diferentes números de componentes, se llegó un a un valor recomendado $L = 8$.

3.2.2.4. Comparación de métodos de suavización

El método de suavización utilizado posee las siguientes características: número de repeticiones (3,5,7,9,10) y número de puntos por repetición (5-50 puntos en intervalos de 5). Cada combinación de método de suavización y método de muestreo (repeticiones y número de puntos) se repitieron 10 veces. Se utilizó el software R.14.0 con la librería KernSmooth para usar los métodos spline, LOESS y Kernel. Las comparaciones se basaron en el indicador $\chi^2_{Tot} = \sum_{i=1}^{3}\chi^2_{red_i}$ (suma de χ^2_{red} para u_i). La Figura 3.2 muestra este indicador para cada combinación etiquetada mediante "Smoothing type". La suavización MAP tiene el error más pequeño en el área de [40-100] puntos.

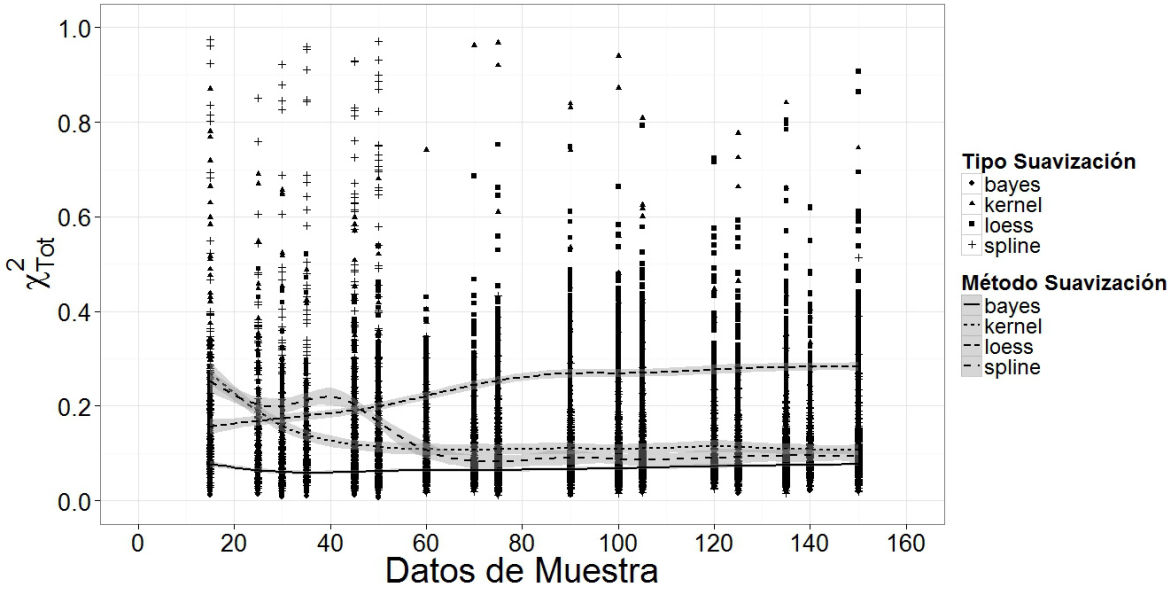


Figura 3.2: χ^2_{Tot} para diferentes tamaños de muestras y tipo de suavización

3.2.3. Aplicación de redes neuronales en tiempo continuo

La metodología propuesta está basada en las derivadas con respecto al tiempo de una función basada en la forma de la red neuronal recurrente mencionada en (3.2). Las etapas del procedimiento fueron las siguientes:

1. A partir de la serie de datos suavizada $m_i(t)$ calcular las derivadas con respecto al tiempo, usando un pequeño intervalo de tiempo, para cada unas de las series de expresión genética. Estas derivadas aproximan los valores de las derivadas temporales reales de los datos de expresión $\dot{\hat{x}}_i(t)$ o $\dot{m}_i(t) \approx \dot{\hat{x}}_i(t)$..

2. Con la función de error (3.1) y los estimados de las derivadas con respecto al tiempo de (3.2) se obtiene una función de error basada en las derivadas con respecto al tiempo de la función suavizada y las derivadas temporales calculadas por la red

neuronal recurrente. La ecuación del error es de la forma:

$$E = \frac{1}{TN}\sum_{t=1}^{T}\sum_{i=1}^{N}[\dot{x}_i(t) - \hat{\dot{x}}_i(t)]^2. \tag{3.17}$$

La ecuación (3.17) es un caso especial de la ecuación de error (3.1). Para aproximar las derivadas con respecto al tiempo se utilizó la expresión de la CTRNN en la siguiente forma:

$$\dot{x}_i(t) = \frac{f\left[\sum_{j=1}^{N} w_{i,j}x_j + \sum_{k=1}^{K} v_{i,k}u_k + \beta_i\right] - \lambda_i x_i}{\tau_i}, \tag{3.18}$$

En este caso se emplea la función $\tanh(x)$ en lugar de la función genérica f. También se omitió el término $\sum_{k=1}^{K} v_{i,k}u_k$ debido a que representa el efecto de otra variable independiente en el modelo. La ecuación reducida es de la forma:

$$\dot{x}_i(t) = \frac{f(\sum_{j=1}^{N} w_{i,j}x_j + \beta_i) - \lambda_i x_i}{\tau_i}, \tag{3.19}$$

que es dependiente en los parámetros $w_{ij,}\beta_i, \lambda_i, \tau_i$. Con la combinación de (3.17) y (3.18) se obtiene

$$E = \frac{1}{TN}\sum_{t=1}^{T}\sum_{i=1}^{N}[\dot{x}_i(t) - \frac{f(\sum_{j=1}^{N} w_{i,j}x_j + \beta_i) - \lambda_i x_i}{\tau_i}]^2 \tag{3.20}$$

Con el uso de (3.10), que tiene un término sigma, la función de error se expresa:

$$E = \frac{\frac{1}{TN}\sum_{t=1}^{T}\sum_{i=1}^{N}[\frac{f(\sum_{j=1}^{N} w_{i,j}x_j+\beta_i)-\lambda_i x_i}{\tau_i} - \dot{\hat{x}}_i(t)]^2}{\sigma} + \frac{1}{2}\ln(2\pi\sigma) \tag{3.21}$$

Esta función de error es dependiente de los parámetros w_{ij}, β_i, λ_i, τ_i las derivadas temporales $\dot{x}_i(t)$ las concentraciones $x_i(t)$, la función de error E se vuelve $E(w_{ij,}\beta_i, \lambda_i, \tau_i, \sigma) \equiv E$. Dado que se usaron los datos suavizados como representación de las concentraciones reales y sus derivadas temporales o $\dot{m}_i(t) \approx \dot{\hat{x}}_i(t)$ la ecuación anterior toma la forma

$$E = \frac{1}{\sigma}\frac{1}{TN}\sum_{t=1}^{T}\sum_{i=1}^{N}[\dot{m}_i(t) - \frac{1}{\tau_i}f(\sum_{j=1}^{N} w_{i,j}m_j + \beta_i) - \lambda_i m_i]^2 + \frac{1}{2}\ln(2\pi\sigma) \tag{3.22}$$

3.2.4. Algoritmo de solución de la red neuronal aplicado

1. Usamos un método de optimización en la función de error $E(w_{ij,}\beta_i, \lambda_i, \tau_i, \sigma)$ basada en el método del gradiente conjugado y se obtuvieron los estimadores $\hat{w_{ij}}$, $\hat{\beta}_i$, $\hat{\lambda}_i$, $\hat{\tau}_i$, $\hat{\sigma}$. Con ello es posible calcular las derivadas temporales del sistema dinámico.

2. Con $\hat{w}_{ij,}\hat{\beta}_i, \hat{\lambda}_i, \hat{\tau}_i, \hat{\sigma}$ y los datos suavizados $m_i(t)$ se estiman las derivadas temporales mediante (3.19). Este procedimiento calcula la derivada temporal para cada punto de la serie suavizada. Con (3.19) y los datos suavizados se tiene

$$\dot{x}_i(t) = \frac{f(\sum_{j=1}^{N} \hat{w}_{i,j} m_j(t) + \hat{\beta}_i) - \hat{\lambda}_i m_i(t))}{\tau_i} \tag{3.23}$$

3. Se realizó la simulación de los niveles de expresión de cada gen aplicando las derivadas estimadas por el modelo en el primer punto real para cada serie de una manera iterativa.

$$\begin{aligned} x_i(t_1 + \Delta t) =& x_i(t_1) + \dot{x}_i(t_1)\Delta t \\ t_2 =& t_1 + \Delta t \\ x_i(t_2 + \Delta t) =& x_i(t_2) + \dot{x}_i(t_2)\Delta t \end{aligned} \tag{3.24}$$

3.3. Resultados obtenidos con diferentes conjuntos de datos

3.3.1. Aplicación sobre datos simulados

La metodología propuesta anteriormente se aplicó en datos sintéticos generados de la siguiente manera:

1. Se inicia con un modelo represilador de tres genes (A,B,C)

2. Se muestrean 20 puntos temporales con 3 repeticiones cada uno en la parte estable del ciclo

3. Se añade un término estocástico en cada punto de muestreo, el término estocástico tiene las características de un ruido gaussiano con media=0 y varianza σ_p^2

La elección del modelo represilador fue para asegurar que se pudieran generar muestras de manera sencilla a fin de tener un conocimiento total del sistema dinámico que las generaba. El método de muestreo utilizado (pocos puntos temporales y repeticiones) se hizo para representar una situación con pocos datos, una situación común en la reconstrucción de las redes de regulación genética. Esta aplicación de la metodología fue repetida 100 veces para asegurar que los resultados fueran estables y representativos.

Los resultados completos del procedimiento en todo el ciclo de simulación se muestra en la Figura 3.3, en esta las líneas representan la media de las trayectorias suavizadas, así como la media de las trayectorias del modelo, las cuales empiezan siempre en el mismo punto temporal pero con un valor diferente de concentración, esto debido a la adición del término estocástico. Observamos que los datos predichos por el modelo de la CTRNN es la fase estable como se observa en la Figura 3.4 concuerda con el comportamiento real de las interacciones del sistema dinámico del represilador, es decir, cuando hay un incremento de la concentración del gen A existe un decremento en la concentración del gen B, y cuando la concentración de B aumenta, C disminuye, esto de acuerdo con las dinámicas reales. Debe de hacerse notar que el procedimiento genera un modelo CTRNN en el cual, a partir de una condición inicial, se reproducen las interacciones seguida por las variables del sistema dinámico. Por asociación directa de los pesos de la interacción gen a gen del modelo estimado, se alcanza una precisión de alrededor del 30 % en el modelo represilador incluso cuando aumenta el número de genes. Esto es consistente con resultados previo de la literatura en el contexto de las redes neuronales artificiales recurrentes [25]. Con estos

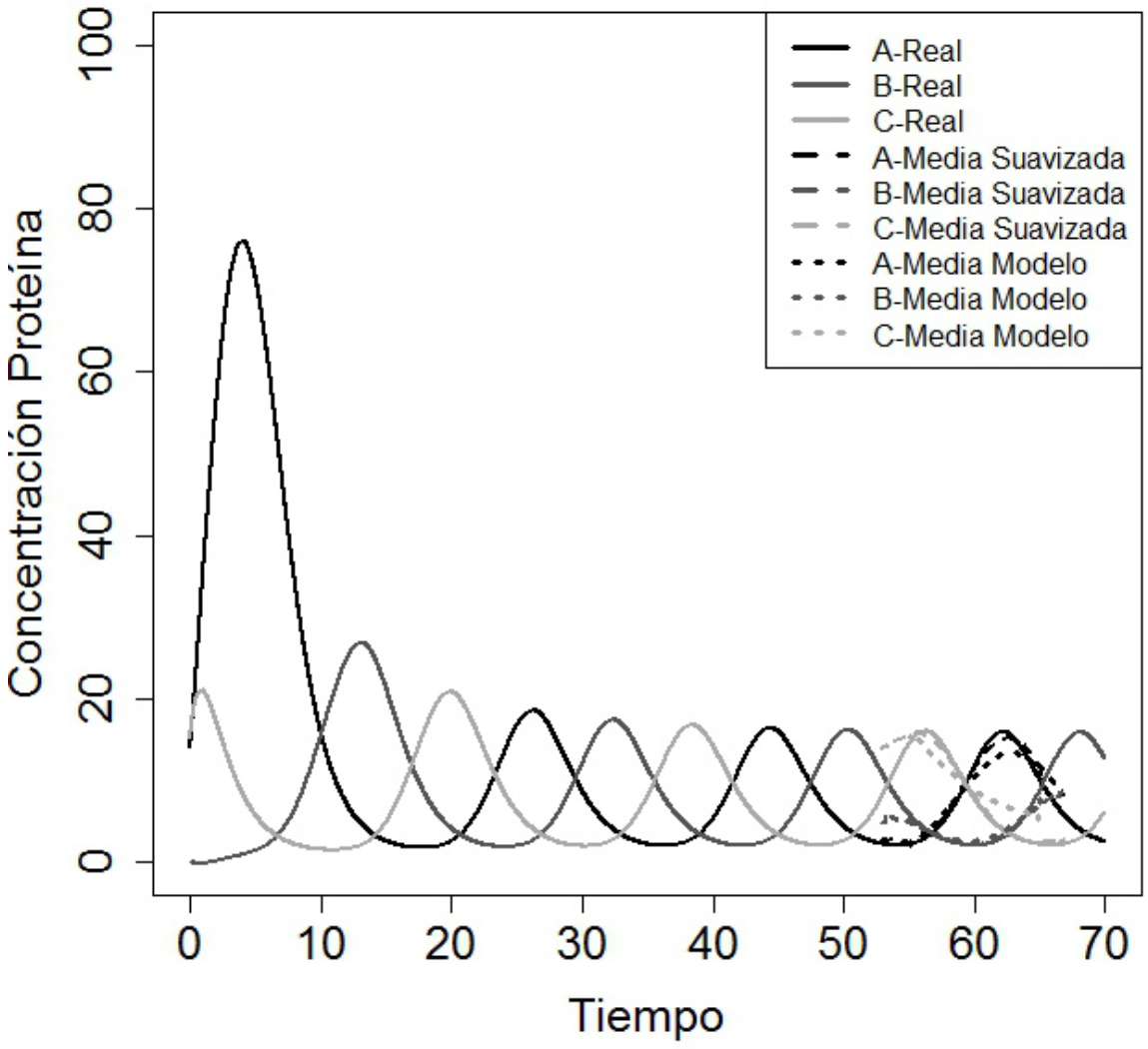


Figura 3.3: Datos sintéticos completos y los resultados del modelo CTRNN en una fase estable

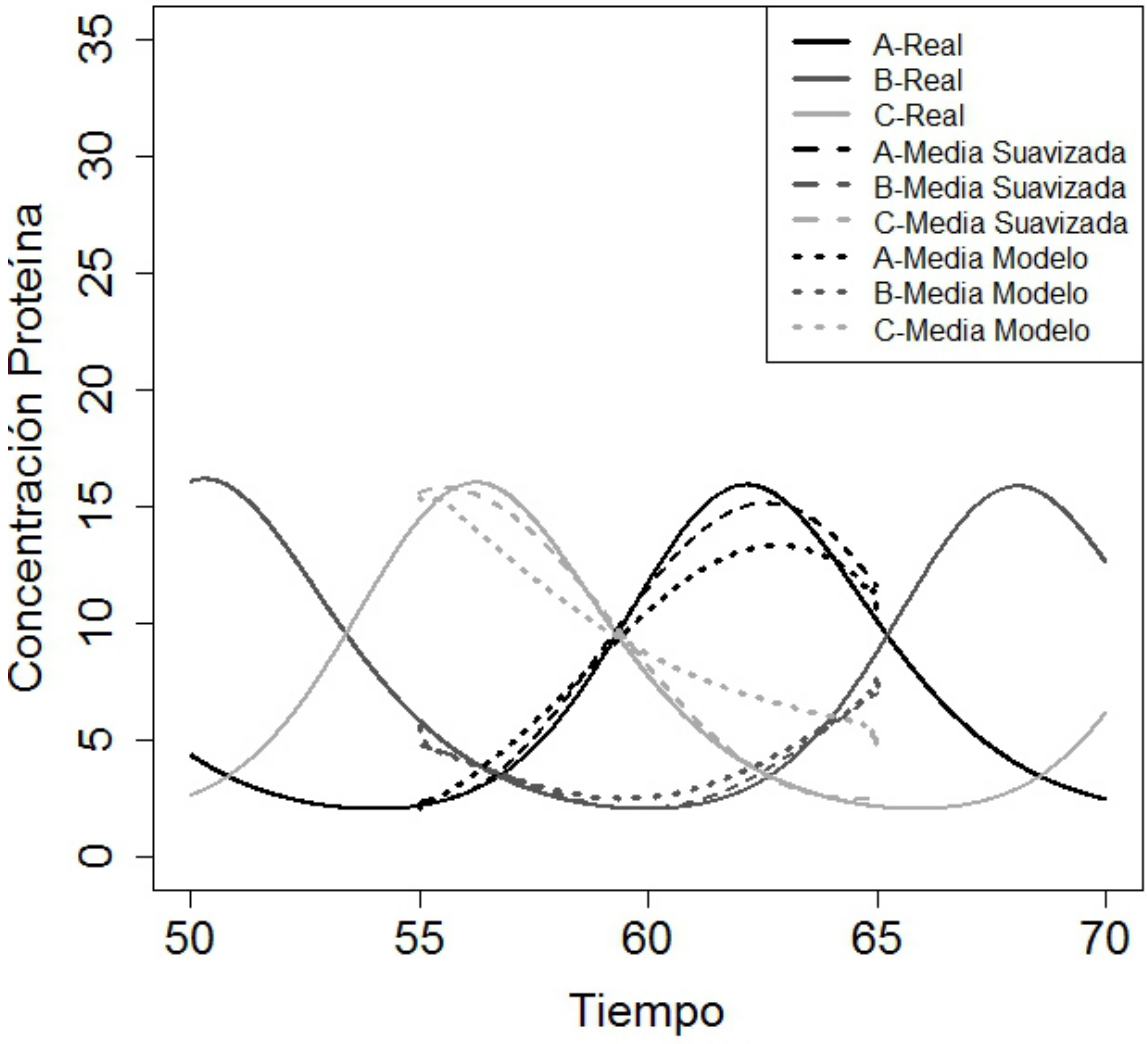


Figura 3.4: Fase estable de los datos sintéticos y los datos del modelo CTRNN

resultados como evidencia, se realizó una comparación de esta metodología con respecto a una alternativa viable para medir su desempeño en términos de ajuste y tiempo de procesamiento.

3.4. Estimación del costo computacional

Este método implica ahorros en el esfuerzo computacional medido en términos de evaluación de las funciones de costo. Para visualizar esto se considero un método de integración de Euler en un intervalo T_0 definido por el primer y último punto observable. El número necesario de evaluaciones de la función de costo para integrar el sistema escala como $\frac{T_0 N}{\triangle t}$, donde $\triangle t$ es un pequeño paso de integración y N corresponde al número de variables. Asumiendo que existen T puntos observables para cada variable, el costo computacional para evaluar la función de error (3.20) crece como $T\frac{T_0 N}{\triangle t}$. En contraste el método propuesto en este trabajo requiere TN evaluaciones de la función de costo para el cálculo de (3.20).

Sin embargo, existe un impacto de dos factores no discutidos en la evaluación de la función de costo: primero, la elección del tamaño del paso para evaluar las funciones de gradiente en el método propuesto, y segundo, el número de términos de derivación que entran en la función de error y que dependen del tamaño de la muestra.

Para obtener un estimado de la importancia de estos factores, se realizó un experimento contra una alternativa viable. El método alternativo se baso en la minimización de (3.1) donde los $\hat{x}_{i,r}(t)$ se obtienen mediante la solución numérica de (3.2) en el lapso de tiempo definido por la muestra utilizando un paso de integración $\triangle t$ y como parámetros un conjunto de w's, v's, β's, τ's y λ's (codificados de ahora en adelante por el vector $\boldsymbol{\theta}$). La minimización de la función de error fue hecha con un algoritmo genético en el vector $\boldsymbol{\theta}$[39].

Inicialmente se empleó un algoritmo basado en búsqueda aleatoria en un intervalo cerrado el cual proporcionó valores muy altos del error cuadrado medio, incluso usando un alto número de iteraciones. Este resultado motivó a la búsqueda de un método alternativo, en este caso, un algoritmo genético. Este algoritmo utiliza intervalos cerrados basados en las soluciones encontradas por la red neuronal recurrente y tiene las siguientes características: generación de población a través de una distribución uniforme continua, recombinación basada en aritmética local y mutación aleatoria uniforme con un factor de elitismo del 5 % con una población inicial de 50 y un límite superior de 50 iteraciones.

El experimento utilizó datos sintéticos del represilador de 3 genes en una fase estable, y se controlaron los siguientes factores: método, número de puntos muestrales, número de repeticiones del experimento, tamaño del paso de integración y diferentes niveles de ruido en las mediciones, todo lo anterior impacta en el error cuadrado medio y tiempo de procesamiento (medido en segundos). La configuración del experimento fue: 20 % de los datos se dejaron fuera para comparación y el restante 80 % se uso para entrenar el modelo. Después de entrenar el modelo, se empleó para predecir el 20 % restante y estos datos fueron comparados contra las mediciones reales mediante el error cuadrático medio.

Los niveles para cada factor son:

1. Método: CTRNN y Algoritmo Genético
2. Número de puntos muestrales: 10, 20 y 30
3. Número de repeticiones del experimento: 1, 2, 3
4. Tamaño de paso de integración: 0.01, 0.05, 0.10, 0.50
5. Niveles de ruido: 10 %, 20 % y 30 % de la varianza de las proteínas,

Para cada combinación de los factores se generaron 5 muestras con las características antes mencionadas, de tal manera que se pudiera hacer un diseño factorial completo. Los métodos fueron implementados en R 3.1.0 y el hardware usado para las pruebas fue un procesador Intel(R) Xeon(R) CPU X5690 3.47GHz. Las medias para el factor método se muestra en la Figura 3.5 donde CTRNN resuelta por el método propuesto tiene una media menor para el error cuadrático medio así como para el tiempo de procesamiento que el algoritmo genético. El valor P asociado con este factor es $p < 0{,}0001$ de tal manera que podemos rechazar la hipótesis nula de que el modelo no afecta a la media. Las tablas completas para el ANOVA del tiempo de procesamiento y el error se encuentran en el apéndice. Estos resultados experimentales muestran buena generalización a pesar del uso de herramientas sencillas de optimización, lo cual confirma la metodología propuesta de mantener separado la fase de suavizado del modelo para de esta manera tener un problema de optimización más simple. La introducción de errores por la suavización son manejados mediante la construcción de una función de error generalizados para los parámetros de suavizado, y se realizaron experimentos preliminares para verificar la calidad del suavizamiento propuesto (ver Figura 3.2)

3.5. Comparación de los resultados con otros datos, modelos y algoritmos de solución

3.5.1. Conjunto de datos

Para contrastar los resultados de la investigación, se toma como punto de referencia el trabajo [20], el único método previo en la literatura que incluye una estrategia de suavización para optimización de parámetros de modelos dinámicos no lineales de Redes

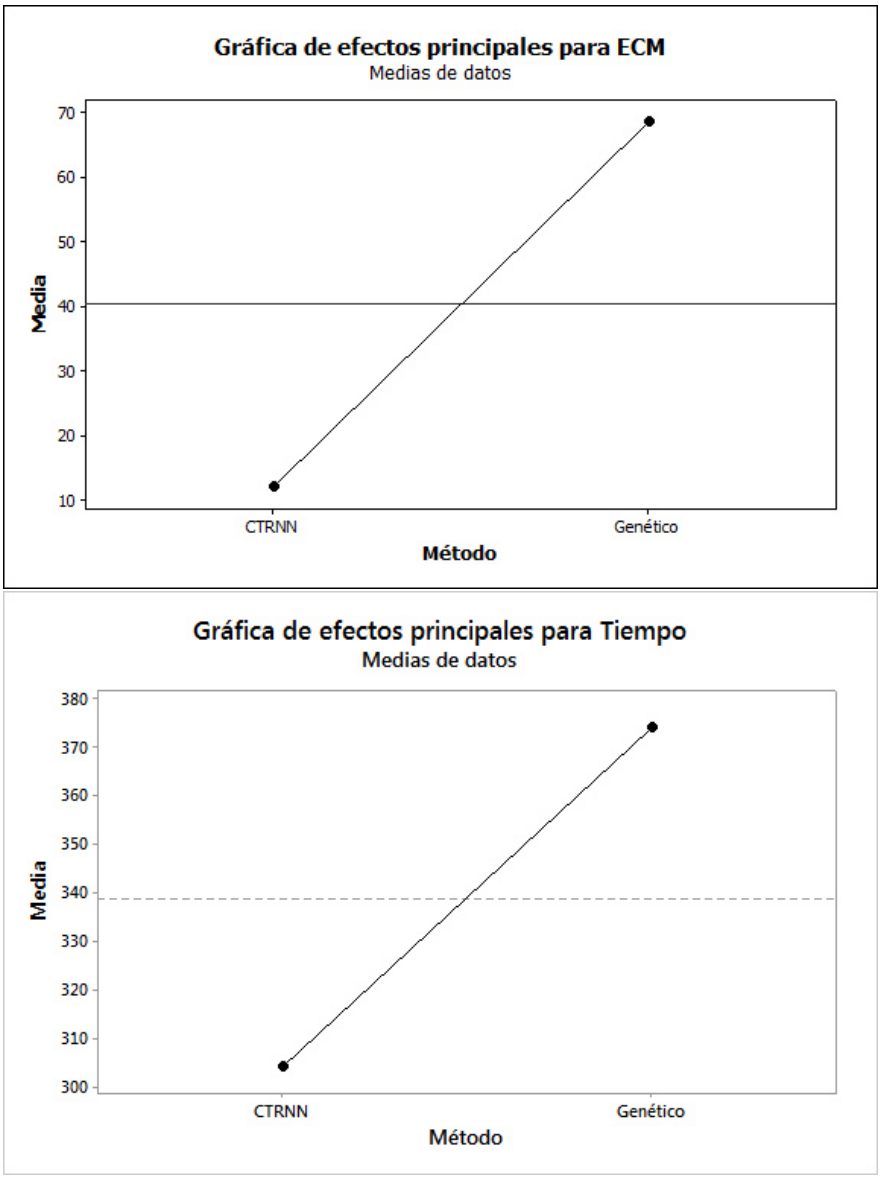


Figura 3.5: Medias del experimento factorial completo para MSE y tiempo de procesamiento

de Regulación Genética. El interés principal de [20] son las interacciones de los genes, sin embargo, sus resultados también generan dinámicas globales que pueden ser comparadas con la metodología propuesta. El caso de estudio de [20] son los datos de expresión de DREAM2 con cinco genes, este conjunto de datos son generados a partir de una red pequeña bio-diseñada, a través de la inserción de nuevas combinaciones promotor/gen en el DNA cromosómico de la levadura; este conjunto de datos fue creado como una base de comparación para la inferencia en redes [42, 43]. Se empleó este conjunto de datos para aplicar la metodología propuesta y compararla con otros métodos. En todos los casos, la estrategia general (especificada en las siguientes subsecciones) consiste en una combinación de un modelo con una técnica de optimización.

3.5.2. Modelos

Cada uno de los métodos en [40] usa un modelo basado en un sistema S el cual se describió en una sección anterior.

3.5.3. Comparación

En el trabajo de [40], el error cuadrático medio (ECM) se uso para comparar diferentes métodos en conjuntos de datos reales. El ECM se ha usado como una medida de ajuste y se define como la diferencia entre los valores de expresión calculados por el modelo y las dinámicas experimentales observadas, entre más pequeño el valor, mejor es la relación entre la dinámica observada y la calculada por el modelo [44]. De acuerdo a [44] esta medida fue introducida por primera vez en esta aplicación por [45] y después usada en los trabajos de [46, 47, 48, 49].

Se aplicó la metodología propuesta en este trabajo para cada uno de los conjuntos de datos y se calculó el ECM para cada uno de ellos, utilizando las siguientes condiciones: el

primer punto experimental en cada conjunto de datos representa el valor inicial a partir del cual las dinámicas de los modelos evolucionan. En [20] no existe una medición explícita del ECM de su modelo con respecto a los datos originales, sin embargo se presenta una comparación gráfica del estimado. Obtuvimos el estimado del ECM a través del programa "g3data" que permite extraer valores numéricos de gráficas con alta precisión [50]. Se tomaron veinte repeticiones para cada uno de los modelos a comparar. Los resultados se muestran en la Figura 3.6 con una comparación de gráfica de caja sobre el ECM para cada uno de ellos.

Para hacer una comparación con datos fuera de muestra el método propuesto se comparo contra un algoritmo genético usando los datos reales de los conjuntos de datos de: Spellman (6 y 7 genes) , DREAM 2 (5 genes) y DREAM7 [51]. Los algoritmos se entrenaron con 80 % de los datos y los datos remanentes se usaron para calcular el ECM de los datos predichos por los algoritmos. Los resultados se muestran en la Figura 3.7 donde se comparan mediante una gráfica de caja del error cuadrado medio para cada conjunto de datos.

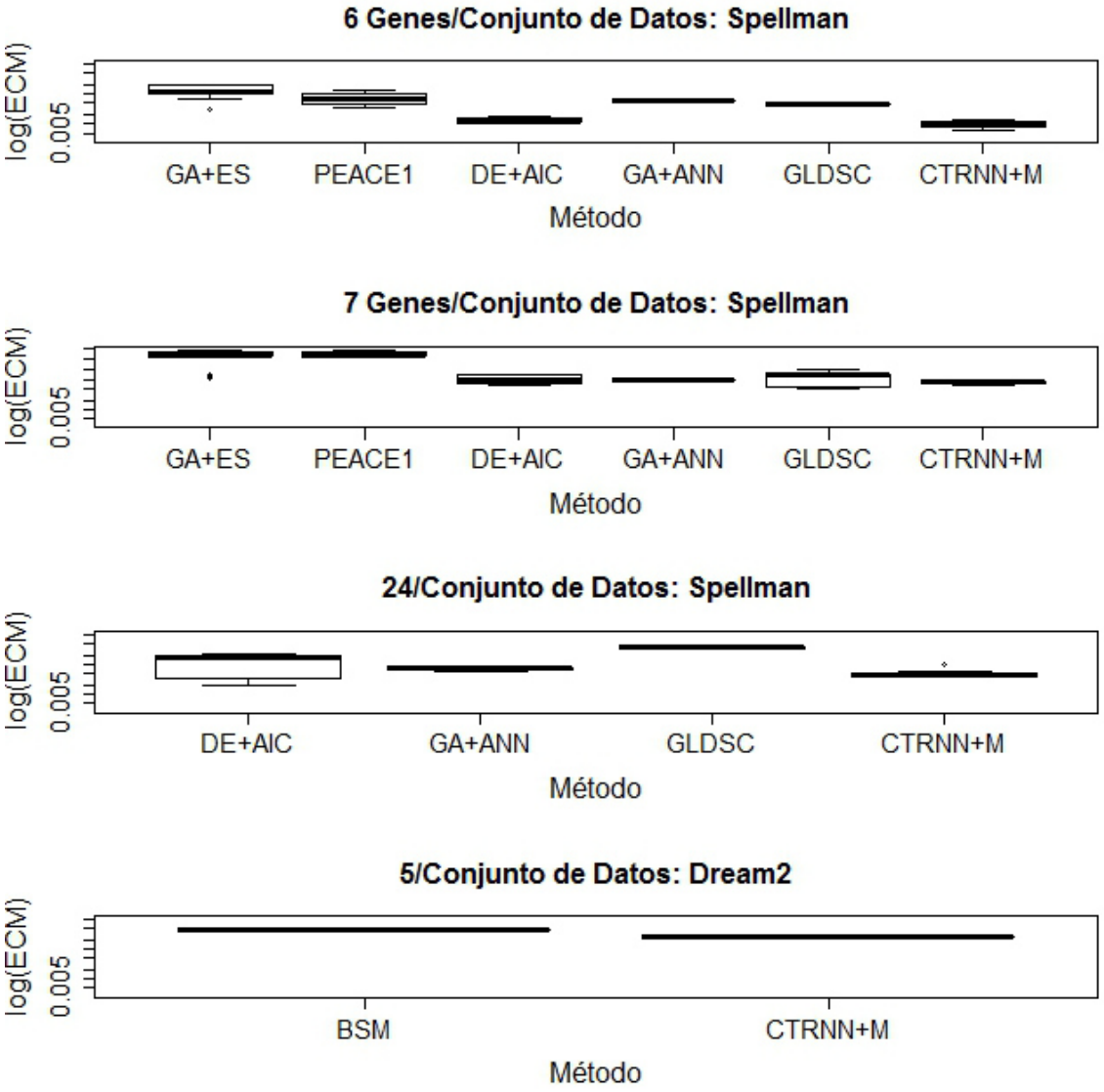


Figura 3.6: Comparación del ECM para diferentes métodos con 6,7 y 24 genes del conjunto de datos de Spellman y el conjunto de datos de 5 genes de DREAM2. La razón (Mejor ECM de método alterno/ECM de CTRNN+M) para cada conjunto de datos son los siguientes: 1.56, 1.14, 1.85, y 1.65. Las pruebas ANOVA para cada conjunto de datos da un $p < 0.0001$, lo que implica una diferencia entre las medias de los métodos.

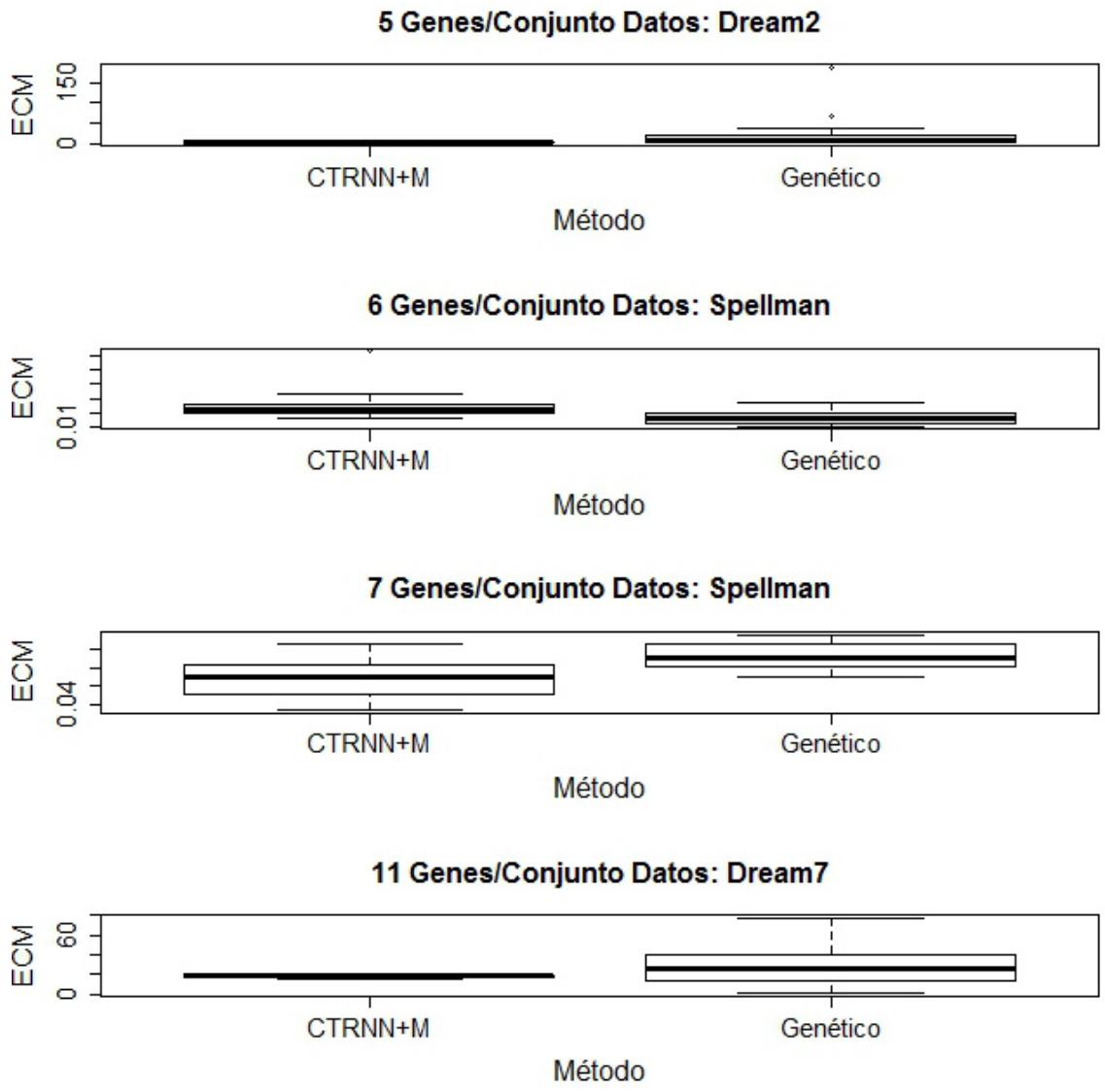


Figura 3.7: Comparación del ECM para diferentes métodos para: conjunto de datos de 5 genes de DREAM2; 6, 7 del conjunto de datos de Spellman y 11 de DREAM7. Las pruebas ANOVA para cada uno dan un valor $p < 0.0001$, lo cual implica que las medias de los métodos son diferentes entre ellos.

Capítulo 4

Conclusiones

En la presente tesis se presentó un modelo basado en redes neuronales en tiempo continuo que permite la recuperación de la dinámica de un sistema de redes de regulación genética. Se comprobó que la estrategia de suavización de los datos junto con la red neuronal permite una ganancia en tiempo de procesamiento así como de disminución del error con respecto a algoritmos usados para la resolución de este tipo de sistemas. Por otra parte al aplicarse esta metodología a conjuntos de datos reales se encontró que el error cuadrado medio obtenido con esta técnica es competitivo con respecto a otros métodos de la literatura, los cuales utilizan diferentes tipos de modelos y algoritmos de solución. Este modelo es lo suficientemente general para ser aplicado a diferentes conjuntos de datos, además de ser adecuado para representar redes de regulación genética, sin embargo la posibilidad de realizar inferencias con respecto a la relación entre los genes es débil, lo cual concuerda con otros métodos y experimentos anteriores.

Capítulo 5

Recomendaciones para Trabajos Futuros

En la investigación futura se explorarán: medidas adicionales de relación entre genes y su relación con un sistema dinámico, la aplicación a diferentes conjuntos de datos de redes de información genética y su posible uso en aplicaciones alternas a las redes de regulación genética.

En este caso las medidas estándar de relación entre genes corresponden a:

Correlación

La medida de relación más común es la correlación lineal, definido por el coeficiente de correlación, el cual toma valores entre 1 y -1, donde un 0 implica que no hay relación lineal entre las variables y un 1 una relación lineal positiva perfecta, y un -1 una relación lineal negativa. El coeficiente de correlación $\rho_{X,Y}$ de Pearson esta definido por (5.1).

$$\rho_{X,Y} = \frac{E\left[(x_1 - \mu_1)(x_2 - \mu_2)\right]}{\sigma_1 \sigma_2} \tag{5.1}$$

Correlación parcial

La correlación parcial es una medida de relación que describe la relación entre dos variables mientras se controla una tercera o más variables. La correlación parcial esta dada por (5.2) donde ρ_{ij} corresponde a la correlación de Pearson.

$$\rho_{ij \cdot k} = \frac{\rho_{ij} - \rho_{ik}\rho_{jk}}{\sqrt{(1 - \rho_{ik}^2)(1 - \rho_{jk}^2)}} \tag{5.2}$$

Estas medidas no toman en cuenta el factor temporal, que dentro de un sistema dinámico es básico para su comprensión. Por ello un trabajo futuro se basará en el estudio de estas medidas de asociación desde un punto de vista dinámico, y ver su derivación a través de un modelo general, como en el planteado en esta tesis.

Aplicación a conjuntos de datos más complejos

El método propuesto se aplicará a conjuntos de datos publicados y de libre acceso más grandes y actuales. Entre los posibles conjuntos de datos se encuentran aquellos correspondientes al DREAM Challenge que se encuentra ya en su iteración número 10, así como los benchmark (referentes de comparación) propuestos por [11] de una manera más detallada y con conjuntos de genes superiores a 15, de tal manera que se pueda medir la utilidad del método donde existan mayor cantidad de posibles relaciones.

Aplicaciones alternas

En la revisión de posibles aplicaciones al método se encontraron similitudes con problemáticas en el ámbito de las series de tiempo financieras, las cuales pueden contener ruido de medición y tener pocas observaciones [53]. Otra área que presenta problemas similares con respecto a la reconstrucción e inferencia de redes corresponde al estudio de redes neuronales biológicas donde las señales eléctricas provenientes de diferentes área del cerebro sirven como base para ver si existe relación entre dichas regiones cuando se realiza alguna función específica [54, 55].

Bibliografía

[1] Chen L, Wang RS, Zhang XS (2009) Biomolecular networks: methods and applications in systems biology. Wiley, Hoboken

[2] Yong Wang. (2013). Gene Regulatory Networks. Encyclopedia of systems biology. (pp. 801-804). Springer Publishing Company, Incorporated.

[3] [Transcripción de proteínas] (2016) note = {http://hnncbiol.blogspot.mx/2008/01/sintesis-de-proteinas_22.html}

[4] Hidde De Jong. Modeling and Simulation of Genetic Regulatory Systems: A Literature Review. RR-4032, 2000. inria-00072606

[5] Jia Meng and Yufei Huang (2013). Gene Regulation. Encyclopedia of systems biology. (pp. 797-800). Springer Publishing Company, Incorporated.

[6] Jürg Bähler and Samuel Marguer (2013). DNA Microarrays. Encyclopedia of systems biology. (pp. 609-610). Springer Publishing Company, Incorporated.

[7] Wang, J. (2008). Computational biology of genome expression and regulation—a review of microarray bioinformatics. Journal of Environmental Pathology, Toxicology and Oncology, 27(3).

[8] Harbison CT, Gordon DB, Lee TI, Rinaldi NJ, Macisaac KD, Danford TW (2004) Transcriptional regulatory code of a eukaryotic genome. Nature 431:99–104

[9] Lee TI, Rinaldi NJ, Robert F, Odom DT, Bar-Joseph Z, Gerber GK (2002) Transcriptional Regulatory Networks in Saccharomyces cerevisiae. Science 298(5594):799–804

[10] Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., ... & Futcher, B. (1998). Comprehensive identification of cell cycle–regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Molecular biology of the cell, 9(12), 3273-3297.

[11] Schaffter, T., Marbach, D., & Floreano, D. (2011). GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods. Bioinformatics, 27(16), 2263-2270.

[12] Hidde De Jong. Modeling and Simulation of Genetic Regulatory Systems: A Literature Review. RR-4032, 2000. inria-00072606

[13] Hecker, M., Lambeck, S., Toepfer, S., Van Someren, E., & Guthke, R. (2009). Gene regulatory network inference: data integration in dynamic models—a review. Biosystems, 96(1), 86-103.

[14] Funahashi, K. I., & Nakamura, Y. (1993). Approximation of dynamical systems by continuous time recurrent neural networks. Neural networks, 6(6), 801-806.

[15] Harvey, I., Husbands, P., & Cliff, D. (1994). Seeing the light: Artificial evolution, real vision (pp. 392-401). Falmer: School of Cognitive and Computing Sciences, University of Sussex.

[16] Quinn, M. (2001). Evolving communication without dedicated communication channels. In Advances in Artificial Life (pp. 357-366). Springer Berlin Heidelberg.

[17] Zhong-Yuan Zhang (2013). Identification of Gene Regulatory Networks, Neural Networks. Encyclopedia of systems biology. (pp. 941-943). Springer Publishing Company, Incorporated.

[18] Mjolsness, E., Castano, R., Mann, T., & Wold, B. (2000). From coexpression to coregulation: an approach to inferring transcriptional regulation among gene classes from large-scale expression data.

[19] van Someren, E. P., Wessels, L. F., & Reinders, M. J. (2001, June). Genetic network models: A comparative study. In BiOS 2001 The International Symposium on Biomedical Optics (pp. 236-247). International Society for Optics and Photonics.

[20] Mazur, J., Ritter, D., Reinelt, G., & Kaderali, L. (2009). Reconstructing nonlinear dynamic models of gene regulation using stochastic sampling. BMC bioinformatics, 10(1), 448

[21] Luıs L. Fonseca, Weiwei Yin, Melissa L. Kemp and Eberhard O. Voit (2013), Metabolic Systems Modeling, Power-Law Functions. (pp. 1272-1275). Springer Publishing Company, Incorporated.

[22] Wang, Y. X., & Huang, H. (2014). Review on statistical methods for gene network reconstruction using expression data. Journal of theoretical biology.

[23] Marbach, D., Prill, R. J., Schaffter, T., Mattiussi, C., Floreano, D., & Stolovitzky, G. (2010). Revealing strengths and weaknesses of methods for gene network inference. Proceedings of the National Academy of Sciences, 107(14), 6286-6291.

[24] Marbach, D., Costello, J. C., Küffner, R., Vega, N. M., Prill, R. J., Camacho, D. M., ... & DREAM5 Consortium. (2012). Wisdom of crowds for robust gene network inference. Nature methods, 9(8), 796-804.

[25] Xu, R., Venayagamoorthy, G. K., & Wunsch II, D. C. (2007). Modeling of gene regulatory networks with hybrid differential evolution and particle swarm optimization. Neural Networks, 20(8), 917-927

[26] Sun, J., Garibaldi, J. M., & Hodgman, C. (2012). Parameter estimation using metaheuristics in systems biology: a comprehensive review. Computational Biology and Bioinformatics, IEEE/ACM Transactions on, 9(1), 185-202.

[27] Varah, J. M. (1982). A spline least squares method for numerical parameter estimation in differential equations. SIAM Journal on Scientific and Statistical Computing, 3(1), 28-46.

[28] D'haeseleer, P., Liang, S., & Somogyi, R. (2000). Genetic network inference: from co-expression clustering to reverse engineering. Bioinformatics, 16(8), 707-726.

[29] Raza, K., Alam, M., & Parveen, R. (2014). Recurrent Neural Network Based Hybrid Model of Gene Regulatory Network. arXiv preprint arXiv:1408.5405.

[30] Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., & Friedman, N. (2003). Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. Nature genetics, 34(2), 166-176.

[31] Lopes, F. M., de Oliveira, E. A., & Cesar, R. M. (2011). Inference of gene regulatory networks from time series by Tsallis entropy. BMC systems biology, 5(1), 61.

[32] Cho, K. H., Choo, S. M., Jung, S. H., Kim, J. R., Choi, H. S., & Kim, J. (2007). Reverse engineering of gene regulatory networks. Systems Biology, IET, 1(3), 149-163.

[33] Vohradsky, J. (2001). Neural model of the genetic network. Journal of Biological Chemistry, 276(39), 36168-36173.

[34] Neal, R. M. (1996). Bayesian Learning for Neural Network (Vol. 118). New York: Springer.

[35] Elowitz, M. B., & Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. Nature, 403(6767), 335-338.

[36] Buse, O., Pérez, R., & Kuznetsov, A. (2010). Dynamical properties of the repressilator model. Physical Review E, 81(6), 066206.

[37] Thomas, S. A., & Jin, Y. (2014). Reconstructing biological gene regulatory networks: where optimization meets big data. Evolutionary Intelligence, 7(1), 29-47.

[38] Lee, J. K. (2011). Statistical bioinformatics: for biomedical and life science researchers. John Wiley & Sons.

[39] Luca Scrucca (2013). GA: A Package for Genetic Algorithms in R. Journal of Statistical Software, 53(4), 1-37. URL http://www.jstatsoft.org/v53/i04/.

[40] Sîrbu, A., Ruskin, H. J., & Crane, M. (2010). Comparison of evolutionary algorithms in gene regulatory network model inference. BMC bioinformatics, 11(1), 59. [Data set].doi:10.1186/1471-2105-11-59

[41] Ruz, G. A., & Goles, E. (2013). Learning gene regulatory networks using the bees algorithm. Neural Computing and Applications, 22(1), 63-70.

[42] Stolovitzky, G., Prill, R. J. and Califano, A. (2009), Lessons from the DREAM2 Challenges. Annals of the New York Academy of Sciences, 1158: 159–195.

[43] Cantone, I., Marucci, L., Iorio, F., Ricci, M. A., Belcastro, V., Bansal, M., ... & Cosma, M. P. (2009). A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches. Cell, 137(1), 172-181.

[44] Kabir, M., Noman, N., & Iba, H. (2010). Reverse engineering gene regulatory network from microarray data using linear time-variant model. BMC bioinformatics, 11(Suppl 1), S56.

[45] Tominaga D, Koga N and Okamoto M (2000): Efficient numerical optimization algorithm based on genetic algorithm for inverse problem. Proceedings of Genetic and Evolutionary Computation Conference 2000, 251–258.

[46] Kimura, S., Ide, K., Kashihara, A., Kano, M., Hatakeyama, M., Masui, R., ... & Konagaya, A. (2005). Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm. Bioinformatics, 21(7), 1154-1163.

[47] Kimura, S., Sonoda, K., Yamane, S., Maeda, H., Matsumura, K., & Hatakeyama, M. (2008). Function approximation approach to the inference of reduced NGnet models of genetic networks. BMC bioinformatics, 9(1), 23.

[48] Noman, N., & Iba, H. (2007). Inferring gene regulatory networks using differential evolution with local search heuristics. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), 4(4), 634-647.

[49] Kikuchi, S., Tominaga, D., Arita, M., Takahashi, K., & Tomita, M. (2003). Dynamic modeling of genetic networks using genetic algorithm and S-system. Bioinformatics, 19(5), 643-650.

[50] Bauer, B., & Reynolds, M. (2008). Recovering data from scanned graphs: Performance of Frantz's g3data software. Behavior research methods, 40(3), 858-868.

[51] Meyer, P., Cokelaer, T., Chandran, D., Kim, K. H., Loh, P. R., Tucker, G., & Saez-Rodriguez, J. (2014). Network topology and parameter estimation: from experimental design methods to gene regulatory network kinetics using a community based approach. BMC systems biology, 8(1), 13.

[52] Berrones, A., Jiménez, E., Alcorta-García, M. A., Almaguer, F. J., & Peña, B. (2015). Parameter inference of general nonlinear dynamical models of gene regulatory networks from small and noisy time series. Neurocomputing.

[53] Giles, C. L., Lawrence, S., & Tsoi, A. C. (2001). Noisy time series prediction using recurrent neural networks and grammatical inference. Machine learning, 44(1-2), 161-183.

[54] Li, Y., Li, X., Ouyang, G., & Guan, X. (2007). Information flow among neural networks with Bayesian estimation. Chinese Science Bulletin, 52(14), 2006-2011.

[55] Gao, L., Sommerlade, L., Coffman, B., Zhang, T., Stephen, J. M., Li, D., ... & Schelter, B. (2015). Granger causal time-dependent source connectivity in the somatosensory network. Scientific reports, 5.

Apéndice 1

Publicación

Neurocomputing ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Parameter inference of general nonlinear dynamical models of gene regulatory networks from small and noisy time series

Arturo Berrones [a,*], Edgar Jiménez [b], María Aracelia Alcorta-García [b], F-Javier Almaguer [b], Brenda Peña [a]

[a] Universidad Autónoma de Nuevo León, Facultad de Ingeniería Mecánica y Eléctrica, Posgrado en Ingeniería de Sistemas, Cd. Universitaria, San Nicolás de los Garza, NL 66450, México

[b] Universidad Autónoma de Nuevo León, Facultad de Ciencias Físico Matemáticas, Posgrado en Ciencias con Orientación en Matemáticas, Universidad Autónoma de Nuevo León Cd. Universitaria, San Nicolás de los Garza, NL 66450, México

ARTICLE INFO

Article history:
Received 24 April 2015
Received in revised form
24 August 2015
Accepted 27 October 2015
Communicated by J. Torres

Keywords:
CTRNN
Genetic regulatory networks
Genetic expression time series
Bayesian inference

ABSTRACT

A new inference approach to general dynamic models of gene regulatory networks (GRN) is introduced. The methodology is based on a Maximum a Posteriori (MAP) smoothing of time series data from which mean field variables of the dynamics are estimated. The interactions are modeled by a Continuous Time Recurrent Neural Network (CTRNN). Parameter estimation of the CTRNN is performed without the need to numerically solve the system of nonlinear differential equations. The method is tested on a benchmark of real genetic networks and displays superior performance, in terms of the mean squared error of the expression dynamics, compared to other formalisms.

1. Introduction

Although Artificial Neural Networks (ANNs) are flexible models capable to represent arbitrary nonlinear dependencies, adequate generalization from insufficient samples often requires computationally intensive techniques like re-sampling or Bayesian training [1]. From a probabilistic standpoint, the Bayesian approach optimally uses the available data in the sense of the expected out of sample error [2,3]. Standard Bayesian training procedures like Markov Chain Monte Carlo sampling require a careful exploration of the posterior distribution associated to the network's parameters, which implies a large number of error function evaluations [1,3]. This aspect might be responsible for the almost absent use of Bayesian strategies in the training of dynamical models like the Continuous Time Recurrent Neural Network (CTRNN).). With these kind of systems, each error function evaluation depends on the numerical integration of a set of coupled nonlinear differential equations, giving a numerical burden that makes even the simple training by direct error function minimization a difficult task [8]. In this contribution we propose an approximate Bayesian training for the CTRNN capable of generalization at affordable computational costs. Our main motivation is gene expression time series, where data is scarce and corrupted by strong measurement noise but generated by complex nonlinear interactions, making generalization an important issue for this domain [4].

In addition to limited and noisy data, the large scale of gene interaction networks makes parameter inference for general nonlinear dynamical models a very difficult problem [4]. The methods proposed to solve this problem tend to focus on a particular application and are not robust [7]. The usual approaches therefore make simplifications regarding the underlying dynamics, hence being suitable only in specific contexts [6]. On the other hand, evidence indicates that the available methods to tackle fully nonlinear dynamical situations, like the CTRNN, are capable to accurately reconstruct the observed time series and the essential interactions [8], making them one of the promising paradigms for this problem [9]. However, as already stated, the parameter inference of these models usually involves the numerical solution of a coupled system of differential equations at every iteration of an optimization method, which limits the size of tractable regulatory networks [4]. Moreover, as the available techniques focus on training via solely error minimization, they are in principle more prone to over-fitting than a Bayesian approach.

In this contribution, we propose a new procedure to deal with the difficulties mentioned above. Our proposal consists of the following steps:

* Corresponding author.
E-mail address: arturo.berronessn@uanl.edu.mx (A. Berrones).

http://dx.doi.org/10.1016/j.neucom.2015.10.095
0925-2312/

(i) Use available data to smooth the time series in the most unbiased manner possible, without any reference to the underlying network (i.e. using a nonparametric description).
(ii) Interpret the smoothed time series as an average from some posterior distribution of the underlying network parameters.
(iii) Construct a regularized expression for parameter inference of a Continuous Time Recurrent Neural Network (CTRNN).

The advantage of the separate time series smoothing from network inference is to avoid the inherent computational burden in evaluating likelihood functions from the iteration of dynamic models. By considering posterior distributions subject to produce a given average trajectory, the network inference can be recast in terms of much more tractable likelihood functions, allowing less biased network parameter estimations. The proposed generic procedure can be applied in many different ways, depending on the available data and the relevant interaction quantities. The basic idea is also general enough to be of interest in other global dynamics inference contexts. The concept is implemented on the CTRNN model using smoothing techniques suitable for gene expression time series. The CTRNN model is among the most general dynamic representations of genetic networks [11], although its training is a quite nontrivial task [8]. For instance [8] the CTRNN model is trained from expression data using a particle swarm optimization method to estimate the network parameters and [12] used a generalized extended Kalman filter for weight update in back-propagation through time to train the CTRNN. We therefore propose to train the CTRNN from smoothed trajectories from which average time derivatives of the variables can be estimated. The numerical effort is in this way substantially reduced. The concept of parameter estimation of dynamic models from smoothed trajectories has been originally proposed [10] by using interpolating cubic splines as the smoothing method. A variation of the original concept more suited to short and noisy time series in the context of a particular model of gene regulatory networks has been proposed [4,5]. To our knowledge, this is the first extensive comparison between smoothing methods for short and noisy real data, including an approximately Bayesian approach, instead of synthetic data as in [13,14]. Moreover, we introduce for the first time the average trajectory strategy in the training of a CTRNN, together with a Bayesian regularization which to our knowledge is also new in the context of CTRNN.

Our approach results are on a quite general dynamical model. Clearly, more restricted models with further assumptions on the nonlinearities or coupled linear differential equations should be more efficient in some cases, but for complex systems like gene regulatory networks, it is in generally not possible to beforehand assume particular interaction structures. Our numerical experiments strongly suggest that our approach displays superior generalization performance compared to restricted models when tested on representative gene expression time series benchmarks.

The reverse engineering gene interaction in a dynamical context can be stated as follows: given time series gene expression data $\hat{x}_i(t)$, a model $x_i(t)$ must be inferred. An error function which depends on the model and the sample is defined by,

$$E=\frac{1}{RTN}\sum_{r=1}^{R}\sum_{t=1}^{T}\sum_{i=1}^{N}[x_i(t)-\hat{x}_{i,r}(t)]^2. \tag{1}$$

The summations are over R samples, T time points and N genes. A model for regulatory networks which is widely accepted to be general enough to realistically capture gene interactions in the cell is given by a continuous time recurrent neural network (CTRNN) [15,16],

$$\tau_i\dot{x}_i=f\left[\sum_{j=1}^{N}w_{ij}x_j+\sum_{k=1}^{K}v_{ik}u_k+\beta_i\right]-\lambda_i x_i, \tag{2}$$

where the quantities w's, v's, β's, τ's and λ's are parameters (hereafter encoded by the vector $\boldsymbol{\theta}$), while $\boldsymbol{x}$ and $\boldsymbol{u}$ represent gene expression levels and external variables, respectively.

The base of our approach is to propose a mean field description for the gene expression levels at fixed time, using a joint probability distribution Q. This function, for a simple initial approach, considers the distribution functions of x_i, u_k denoted as q_i and q_k respectively as independent, which gives the expression

$$Q(\boldsymbol{x},\boldsymbol{u})=\prod_{i=1}^{N}q_i(x_i)\prod_{k=1}^{K}q_k(u_k), \tag{3}$$

with

$$\int x_i q_i(x_i)\,dx_i=m_i \tag{4}$$

$$\int u_k q_k(u_k)\,du_k=\overline{u}_k \tag{5}$$

The m's and $\overline{u}$'s give a mean field description of the dynamics.

Using (3) we can write (2) as

$$\tau_i<\dot{x}_i>_Q=f\left[\sum_{j=1}^{N}w_{i,j}m_j+\sum_{k=1}^{K}v_{i,k}\overline{u}_k+\beta_i\right]-\lambda_i m_i \tag{6}$$

Eq. (3) is based on the assumption that the species are statistically independent at a fixed time. This leads to the averaged dynamic Eq. (6). This is analogous to the mean field approach of statistical mechanics: the distribution Q encodes the aggregated effect on each species from all the species at previous times. The averaged dynamic Eq. (6) is therefore associated to the smoothened trajectory rather than to the actual trajectory.

Then a suitable likelihood function that take advantage of the smoothness of the mean field variables can then be defined, for instance in terms of the error, using a subcase of (1) for $R=1$ and (6):

$$V(\boldsymbol{\theta}|\{\dot{\mathbf{m}}\})=\frac{1}{TN}\sum_{t=1}^{T}\sum_{i=1}^{N}[<\dot{x}_i(t)>_Q-\dot{m}_i(t)]^2, \tag{7}$$

where $<\ >_Q$ denotes average under probability function Q.

The posterior density for the network parameters $\boldsymbol{\theta}$ is given by [17],

$$P(\boldsymbol{\theta}|\{\dot{\boldsymbol{m}}\})=\frac{1}{Z}g(\boldsymbol{\theta})h(V), \tag{8}$$

where g is a prior density, h a likelihood density and Z a normalization factor. Several methods can be considered for the extraction of useful information about the network parameters from (8). Consider for instance a Gaussian model with independent priors,

$$h(V)=\frac{1}{\sqrt{2\pi\sigma_\varepsilon^2}}\exp\left(-\frac{\varepsilon^2}{2\sigma_\varepsilon}\right),$$

$$P(\boldsymbol{\theta}|\{\dot{m}\})=h(V)\prod_{r=1}^{R}\frac{1}{\sqrt{2\pi\sigma_r^2}}\exp\left(-\frac{(\theta_r-\mu_r)^2}{2\pi\sigma_r^2}\right) \tag{9}$$

in which case

$$-\ln(P)=\frac{1}{2\sigma_\varepsilon}V+\frac{1}{2}\ln(2\pi\sigma_\varepsilon^2)+\sum_{r=1}^{R}\left\{\frac{1}{2\pi\sigma_r^2}(\theta_r-\mu_r)^2+\frac{1}{2}\pi\sigma_r^2\right\} \tag{10}$$

where $\varepsilon=\sqrt{V}$. From (9) and (10) the underlying parameters can be estimated in a straightforward manner. The MAP estimate, for instance, is directly obtained by the minimization of Eq. (10) with respect to the θ's, μ's and σ's. A probabilistic model which associates a mean value and a variance for the interaction network's parameters follows. This is expected to be more adequate for the reverse engineering of gene interaction that a simple point estimate.

Please cite this article as: A. Berrones, et al., Parameter inference of general nonlinear dynamical models of gene regulatory networks from small and noisy time series, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.10.095

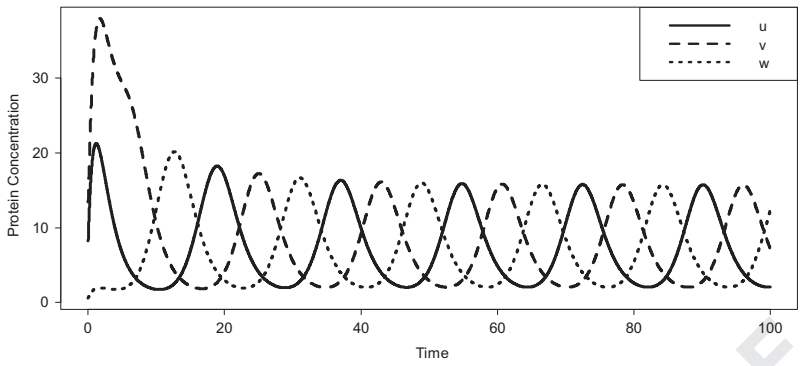


Fig. 1. Concentrations of proteins u_i (u, v and w).

2. Methods

The method previously proposed relies on getting the best possible smoothing trajectory and then applying the minimization of the squared error to get the parameters of the CTRNN. In this section we described our study on a synthetic data set, described in Section 2.1, which we will use to compare different smoothing methodologies with different data sizes and sampling methods. The basis for comparison, methods and results are described in Section 2.2.

These subsections describe the next scheme:

1. Generate data from a known gene regulatory network (synthetic data)
2. Sample data from the simulation
3. Smooth the series with different methods
4. Make a comparison of smoothing methodologies and sampling methods

With these steps we estimated the impact of sampling method and smoothing techniques on synthetic data to get a recommended sample size and smoothing method.

In Section 2.3, the previous results described the method to train the CTRNN with a regularized error function (1) and then apply it to the synthetic data set. In the next subsection, we discuss the estimated computational cost and describe the results of an experiment for comparison against another algorithm, using different factors to control variability of the result.

2.1. Synthetic data

To generate the data we used the Repressilator regulatory network model [18] in its version described in [19]. The concentrations of proteins in the system are given by:

$$\begin{cases} i<3 & \dfrac{dm_i}{dt} = -m_i + \dfrac{\alpha}{1+u_{i+1}^n} + \alpha_0 \\ i=3 & \dfrac{dm_i}{dt} = -m_i + \dfrac{\alpha}{1+u_1^n} + \alpha_0 \end{cases} \qquad i=1,2,3 \dfrac{du_i}{dt} = -\beta(u_i - m_i) \tag{11}$$

where u_i are proportional to protein concentrations while m_i are proportional to the concentrations of mRNA corresponding to those proteins. The system shows a cyclic behavior with a period of stabilization from the initial conditions as seen on Fig. 1.

2.2. Impact of sample size and smoothing method

Sampling of biological data on genetic regulatory networks is usually done on regular intervals, sometimes with repetitions and also has measurement error. To represent this error inherent to experimental conditions, we added a stochastic term on each sampling point distributed as a Gaussian error with mean=0 and variance σ_p^2, where σ_p^2 is the variance of the process for each protein, a process commonly used when artificial data is used to approximate to "real" biological data [20], this variance took the value of 10% the variance of the protein concentration. Sample size is of special importance because each measurement has an associated cost, which is usually high for genetic expression profiles [21]. The compared smoothing methods are: Smoothing spline, LOESS, Kernel and MAP smoothing based on Fourier series, this method is described in next subsection.

To assess the impact of sample size and smoothing method we used the indicator

$$\chi^2_{red} = \frac{1}{n}\sum_{i=1}^{n} \frac{(O-E)^2}{\sigma_p^2} \tag{12}$$

where O is the true value of the synthetic series, E the smoothed value, σ_p^2 is the variance of the process and n is the number of data points.

2.2.1. MAP smoothing

This proposed smoothing method is based on a Fourier series of the form $f(t,\hat{a},L) = \sum_{l=1}^{L} a_l \cos(l(2\pi/T)t)$ where the estimation of the parameters a_l is approximately Bayesian. We considered that parameters a_l are probabilistic so

$$P(a_l) \to P(f_t) \Rightarrow \langle f \rangle = \int f(t) P(f_t)\, df_t, \tag{13}$$

with an associated variance

$$\sigma_t^2 = \langle f^2 \rangle - \langle f \rangle^2. \tag{14}$$

Assuming a sample M and uniform prior, we can propose the posterior probability function

$$P(\vec{a}_l | M) = \frac{1}{z} h(\mu | \vec{a}_l), \tag{15}$$

where h is a likelihood density and z is a normalization factor. The likelihood density function is

$$h(\mu | \vec{a}_l) = \prod_{m=1}^{\mu} N[f(\hat{a}_l), \sigma_m^2] \tag{16}$$

which leads to

$$\ln P = -\frac{1}{\sigma^2}\sum_{m=1}^{M} [\hat{y_m} - f(\hat{a}_l)]^2 - \frac{\mu}{2}(2\pi\sigma_m^2) \tag{17}$$

We minimized the expression (17) by a conjugated gradient method. By systematically considering different number of components we have arrived to a recommended value of $L=8$.

Please cite this article as: A. Berrones, et al., Parameter inference of general nonlinear dynamical models of gene regulatory networks from small and noisy time series, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.10.095

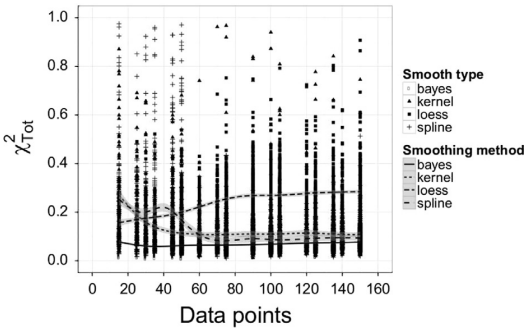


Fig. 2. $\chi_{Tot}{}^2$ for different samples and smoothing methods.

2.2.2. Comparison of smoothing methods

The three factors we used for comparison were:

1. Sampling-Number of repetitions of the sampling method: 3,5,7,9,10 repetitions.
2. Sampling-Number of data points per repetition (5–50 data points by intervals of five).
3. Smoothing method: Spline, LOESS and Kernel methods and the MAP smoothing previously described. We used R.3.1.0 with the library KernSmooth for the first three.

The comparisons are based on the indicator

$$\chi^2_{Tot} = \sum_{i=1}^{3} \chi^2_{red_i} \tag{18}$$

where $\chi^2_{red_i}$ is of the form of the expression (12).

Every combination of smoothing method, sampling method (repetition and number of data points) was repeated 100 times.

2.2.3. Results of comparison of smoothing methods

Fig. 2 shows $\chi_{Tot}{}^2$ (Y-Axis) for each combination of sampling size (X-Axis) and smoothing method tagged by "Smoothing type". A line tagged "Smoothed method" resumes the results, this line is calculated by a smoothing of all the points for each method. The MAP smoothing (bayes) has the smallest in all the X-axis which is related to sampling size.

2.3. Application of the methodology to synthetic data

2.3.1. Description of the methodology

The methodology is based on time derivatives of the recurrent neural network of the form of (2). Steps we followed are:

1. We used the smoothed data series $m_i(t)$ to calculate time derivatives using a small interval of time for each one of times series expression data. We used these calculated time derivatives as a proxy of the real time derivatives $\dot{\hat{x}}_i(t)$ of the expression data, or $\dot{m}_i(t) \approx \dot{\hat{x}}_i(t)$.
2. Using the error function of (1) and the estimation of the time derivative with (2) we created an error function based on the time derivatives of the smoothed data and the time derivatives calculated by the CTRNN. The equation of the error is of the form:

$$E = \frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} [\dot{x}_i(t) - \dot{\hat{x}}_i(t)]^2, \tag{19}$$

which is a sub-case of (1). To approximate the time derivatives we use the CTRNN which expressed for them give:

$$\dot{x}_i(t) = \frac{f\left[\sum_{j=1}^{N} w_{i,j}x_j + \sum_{k=1}^{K} v_{i,k}u_k + \beta_i\right] - \lambda_i x_i}{\tau_i}, \tag{20}$$

In this case we used a $\tanh(x)$ as f. We also omitted the term $\sum_{k=1}^{K} v_{i,k}u_k$ because it represents the effect of another independent variable in the model. The reduced form is

$$\dot{x}_i(t) = \frac{f(\sum_{j=1}^{N} w_{i,j}x_j + \beta_i) - \lambda_i x_i}{\tau_i}, \tag{21}$$

which is dependent on the parameters $w_{ij}, \beta_i, \lambda_i, \tau_i$. Combining (19) and (20) we get

$$E = \frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} \left[\dot{x}_i(t) - \frac{f(\sum_{j=1}^{N} w_{i,j}x_j + \beta_i) - \lambda_i x_i}{\tau_i}\right]^2 \tag{22}$$

By (10), the regularized error function becomes:

$$E = \frac{\frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} \left[\frac{f(\sum_{j=1}^{N} w_{i,j}x_j + \beta_i) - \lambda_i x_i}{\tau_i} - \dot{\hat{x}}_i(t)\right]^2}{\sigma} + \frac{1}{2}\ln(2\pi\sigma) \tag{23}$$

Notice that $E = E(w_{ij}, \beta_i, \lambda_i, \tau_i, \sigma)$. we are using the smoothed data as a proxy for the real concentrations and time derivatives or $\dot{m}_i(t) \approx \dot{\hat{x}}_i(t)$, (23) becomes

$$E = \frac{\frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} \left[\frac{f(\sum_{j=1}^{N} w_{i,j}m_j + \beta_i) - \lambda_i m_i}{\tau_i} - \dot{m}_i(t)\right]^2}{\sigma} + \frac{1}{2}\ln(2\pi\sigma) \tag{24}$$

3. Next we used an optimization method on the error function $E(w_{ij}, \beta_i, \lambda_i, \tau_i, \sigma)$ based on the conjugated gradient and we got estimates $\hat{w}_{ij}$, $\hat{\beta}_i$, $\hat{\lambda}_i$, $\hat{\tau}_i$, $\hat{\sigma}$. These are the base to calculate the time derivatives of the dynamical system.
4. With $\hat{w}_{ij}\hat{\beta}_i, \hat{\lambda}_i, \hat{\tau}_i, \hat{\sigma}$ and the smoothed data $m_i(t)$ we estimate the time derivatives using (21). The procedure calculates the time derivative for each time point of the smoothed series. Using (21) and the smoothed data

$$\dot{x}_i(t) = \frac{f(\sum_{j=1}^{N} \hat{w}_{i,j}m_j(t) + \hat{\beta}_i) - \hat{\lambda}_i m_i(t))}{\tau_i} \tag{25}$$

5. We did the simulation of expression of each gene with the first real data point for each series as starting conditions,

$$x_i(t_1 + \Delta t) = x_i(t_1) + \dot{x}_i(t_1)\Delta t t_2 = t_1 + \Delta t$$
$$x_i(t_2 + \Delta t) = x_i(t_2) + \dot{x}_i(t_2)\Delta t \tag{26}$$

2.3.2. Results of the methodology on synthetic data

We used the proposed methodology on synthetic data generated with the next steps:

1. Create a repressilator model of three genes (A,B,C).
2. Sample 20 time points with 3 repetitions each on the stable part of the cycle.
3. Add a stochastic term on each sampling point distributed as a Gaussian error with mean=0 and variance $\sigma_p{}^2$.

The choice of a simple repressilator model was to ensure we could generate samples easily for study and had a total knowledge of the behavior of the dynamic system. In the case of our chosen sampling method (few time points and repetitions) it was done to represent a data scarcity situation in which a main problem is the reconstruction

Please cite this article as: A. Berrones, et al., Parameter inference of general nonlinear dynamical models of gene regulatory networks from small and noisy time series, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.10.095

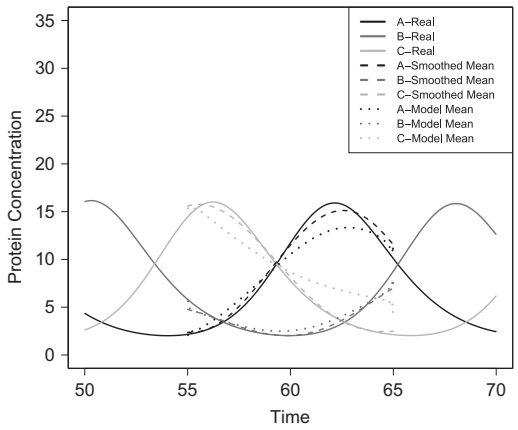


Fig. 3. Stable phase of synthetic data and the results of the CTRNN data.

of genetic regulatory networks. This set up was repeated 100 times to ensure the results were stable and representative.

The results for the procedure in all the simulation cycle are shown in Fig. 3. In which lines represent the mean of smoothed data trajectories, and mean of model trajectories which begins at the same time point but with a different value of concentration, because of the addition of the stochastic term. We observed the data predicted by the model of the CTRNN on the stable phase as seen in Fig. 3. It matches the real behavior of the interactions of the dynamical system of the repressilator, when there is an increase of concentration on gene A there is a decrease on the concentration of B, and when the concentration of B rises C diminishes, in accordance with the actual dynamics. It should be stressed that the procedure generates a CTRNN model, in which, from a given initial condition reproduces the interactions followed by the variables of the actual dynamical system. By direct association of the estimated model's weights with the gene–gene interactions, a precision of around 30% in the repressilator model with increasing number of genes is obtained. This is consistent with previous results found in the literature based on recurrent ANN's [8]. Using these results as evidence, we compared this methodology against a suitable alternative to measure its performance in terms of fit and processing time.

2.4. Estimated computational cost and comparison

Our method implies savings on computational effort measured in terms of cost function evaluations. To see this suppose an Euler integration method on an interval T_0 defined by first and last observed points. The necessary number of cost function evaluations for integrate the system scales like $T_0N/\triangle t$, where $\triangle t$ is an small integration step and N is the number of variables. Assuming T observed points for each variable, the total computational effort to evaluate error function (22) grows like $T(T_0N/\triangle t)$. Our method in contrast, requires TN evaluations of the cost function for the calculation of (22).

However, there is also an impact of two factors not discussed in the evaluation of cost function: first, the choice of step time used to evaluate the gradient functions in the proposed method, and second, the number of derivative terms which enter the error function and depend on sample size.

To get an estimate of the importance of these factors, we did an experiment against a suitable alternative. The alternative method was based on the minimization of (1) where the $\hat{x}_{i,r}(t)$ are obtained by the numerical solution of (2) on the time lapse defined by the sample using an integration step $\triangle t$ using a set of w's, ν's, β's, τ's and λ's as parameters (hereafter encoded by the vector $\boldsymbol{\theta}$). The minimization of the error function was done with a genetic algorithm on the vector $\boldsymbol{\theta}$ [22]. Initially, we used an algorithm based on a random search on a bounded interval which yielded very high MSE even using a high number of iterations, that result made us look for an alternative method. In this case, a genetic algorithm. This algorithm used bounded intervals based on the solutions found by the CTRNN with: random uniform population generation, local arithmetic crossover and uniform random mutation and an elitism factor of 5% with an initial population of 50 and an upper limit of 50 iterations.

The experiment was done using synthetic data of the repressilator system on a stable phase, and accounted for the following factors: method, number of sample points, number of repetitions of the experiment, size integration step and different levels of noise on the measurements, which affect MSE and processing time (measured in seconds). The setup was: 20% of the data was taken out for comparison and the other 80% to train the model. After training the model we used it to predict the 20% left and this data was compared using MSE against the original measurements.

The levels for each factor are

1. Method: CTRNN and Genetic Method.
2. Number of sample points: 10, 20, 30.
3. Number of repetitions of the experiment: 1, 2, 3.
4. Size Integration Step: 0.01, 0.05, 0.10, 0.50.
5. Levels of noise: 10%, 20% and 30% of variance of proteins.

For each combination of factors we generated 5 samples with the sampling characteristics previously mentioned, so we could make a full factorial design. The methods were implemented in R 3.1.0 and the hardware used to run the tests was a Intel(R) Xeon(R) CPU X5690 3.47 GHz. The means for the factor model is shown in Fig. 4 where the CTRNN solved by the proposed method has a lower mean for MSE and processing time than the genetic algorithm. The P-Value associated with this factor is $p < 0.0001$ so we can reject the null hypothesis of the model not affecting the mean. The complete tables for ANOVA of the factorial experiment for MSE and time can be found in the appendix. Experimental results indicate good generalization despite the use of straight forward deterministic optimization tools, which confirms our proposal to kept separated the smoothing phase from the model in order to have a more simple final optimization problem. Possible errors introduced by the smoothing are handled by the construction of a regularized error function for the smoothing parameters, and preliminary experiments on the proposed regularized approach are performed (see Fig. 2) to assess the quality of the introduced level of smoothing.

3. Results: application on real expression data

In the next section, we applied the methodology to different data sets and compare it to results from different methods. In Section 3.1, we described the origins of the data sets, and in 3.2, the models proposed to explain a genetic regulatory network in previous works.

3.1. Data sets

In [23] there is a comparison of different methods to recreate the expression data from the Saccharomyces Cerevisiae cell cycle taken from the Spellman data set included and referenced in this publication. There are three different data sets with different number of genes (6, 7 and 24). As acknowledged by several authors, the work presented in [23] is currently one of the most comprehensive comparative studies regarding artificial intelligence techniques applied to nonlinear dynamical models of GRN [9,24]. Moreover, in [20] the work of [23] is considered a detailed review of available reconstruction algorithms which also makes comparisons of the results of the algorithms among different data sets. It is also considered testing

Please cite this article as: A. Berrones, et al., Parameter inference of general nonlinear dynamical models of gene regulatory networks from small and noisy time series, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.10.095

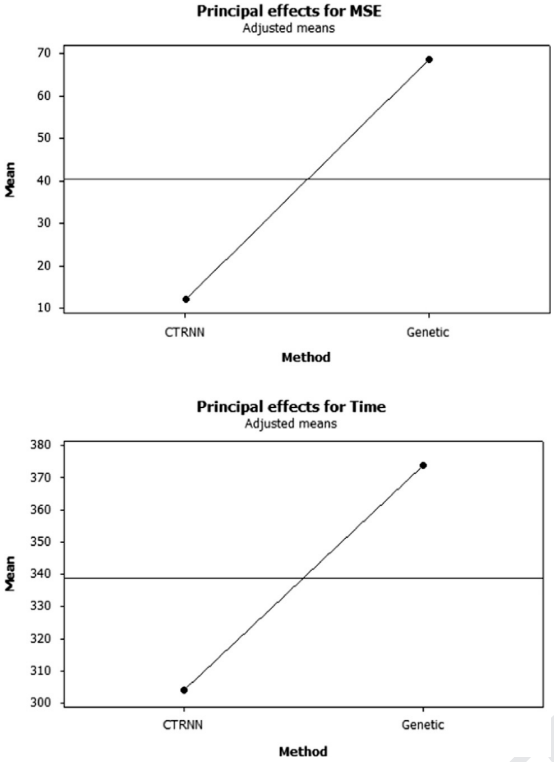


Fig. 4. Means of the full factorial experiment for MSE and processing time.

the work presented in [4], because to our knowledge it is the only previous method which includes a smoothing strategy for the parameter optimization of nonlinear dynamical models of GRN. The main interest in the work presented in [4] is on gene interactions. However, their results also give a global dynamics which can be compared to our methodology. The case study in [4], is the expression data from the DREAM2 challenge 5 gene data set, which is generated from a small bio-engineered network made by inserting new promoter/gene combinations in the chromosomal DNA of budding yeast. This data set was created as a benchmark for comparison of network inference [25,26]. We used these data sets to apply our previously mentioned methodology, and compare against the different methods. In all cases, the general strategy (which we specify in the next subsections) consists on a combination of a model with an optimization technique.

3.2. Models

Each of the methods in [23] used a model based on a S system to represent the interactions which is defined by

$$\frac{dx_i}{dt} = \alpha_i \prod_{j=1}^{n} x_j^{g_{ij}} - \beta_i \prod_{j=1}^{n} x_j^{h_{ij}} \tag{27}$$

where α_i and β_i, the rate constants, represent the basal synthesis and degradation rate, and g_{ij} and h_{ij}, which indicate the strength of the influence of gene j on the synthesis and degradation of the product of gene i, are the kinetic orders.

In [4] the model is defined by

$$\frac{\partial x_i}{\partial t}(t) = s_i - \gamma_i x_i(t) + \sum_{j=1}^{n} |\beta_{ij}| f_{ij}(x_j(t)) \tag{28}$$

where $x_i(t)$ is the concentration of gene i at time t, s_i and γ_i are basal synthesis and degradation rates for each gene i. Variable β_{ij} denotes the regulation strength of component x_j on x_i and f_{ij} is the corresponding regulation function. $\beta_{ij} > 0$ corresponds to an activation, $\beta_{ij} < 0$ to an inhibition, and $\beta_{ij} = 0$ means that there is no regulation from gene j to gene i.

In our framework, the model was the already stated CTRNN.

3.3. Optimization methods

In [23], different optimization methods are used to calculate $\alpha_i, \beta_i, g_{ij}, h$: nesting a genetic algorithm with an evolution strategy (GA + ES), an iterative algorithm based on genetic algorithms (PEACE1), a differential evolution as a search strategy (DE + AIC), an artificial neural network as a model and genetic algorithm for parameter inference (GA+ANN) and a genetic local search (GLDSC). In [4], the method was a Bayesian framework sampled with a Markov chain Montecarlo (BSM). We used a conjugated gradient applied to the error function previously described obtained with the smoothing procedure (M).

3.4. Comparison

In the original work of [23], mean square error (MSE) is used to compare their different methods on real data sets. MSE has been used as a measure of the fitness which is the difference between the calculated expression levels and the observed experimental dynamics, the smaller the value the better the match between the observed dynamics and the one calculated by the model [27]. According to [27] this measure was first introduced in this application by [28] and then used in the works of [29–32] in the same venue.

We applied our methodology to each of the data sets and calculated the MSE for each one, following the same conditions. The first experimental point on each data set is taken as an initial value, from which each of the dynamic models evolve. In [4], there is no explicit measurement of the MSE of their model against the original data, but they give a graphical comparison of the estimated global dynamics. We have been able to get an estimate of their MSE by using the program "g3data", which allows the extraction of numerical values from graphical figures with high precision [33]. Comparison with competing models, we made 20 repetitions of our procedure on each data set. This number was selected because [23] mentioned it as the number of repetitions made by them for those algorithms and models. These results are shown in Fig. 5, compared with a box-plot of the MSE for each of them.

In order to make a comparison with out of sample data, the proposed method was tested against a genetic algorithm on the real data sets of Spellman (6 and 7 genes), DREAM 2 (5 genes) and DREAM7 (11 genes) [34]. The algorithms were trained with 80% of the data and the remaining data as a measure to get the MSE of the predicted data. These results are shown in Fig. 6, compared with a box-plot of the MSE for each of them.

4. Conclusion

We have introduced a new approximately Bayesian training procedure for CTRNN models for coupled nonlinear dynamical systems. Evidence on synthetic and real time series from gene regulatory networks, indicates that despite the high flexibility of the CTRNN model, adequate generalization follows from our approach. The method is competitive with respect to widely accepted dynamic reconstruction from genetic expression time series, based on the evidence of the ratio of MSE of competing methods against MSE of CTRNN + M, which is always higher for other methods and also CTRNN + M has a smaller variance (see Figs. 5 and 6). This is achieved by introducing in the context of CTRNN's training a smoothing phase by which it is

Please cite this article as: A. Berrones, et al., Parameter inference of general nonlinear dynamical models of gene regulatory networks from small and noisy time series, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.10.095

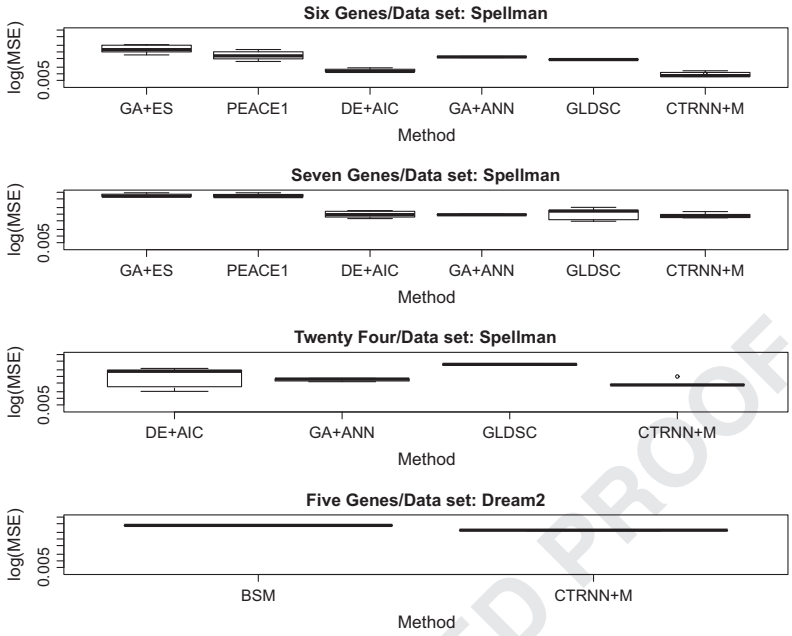


Fig. 5. Comparison of MSE for different methods for 6, 7, and 24 genes from Spellman data set and 5 gene data set from DREAM2. The ratios (Competing methods best MSE)/(MSE of CTRNN+M) for each data set are the following : 1.56, 1.14, 1.85, and 1.65. ANOVA tests for each one gives a $p < 0.0001$ for each one, which means that the means of the methods are different between them.

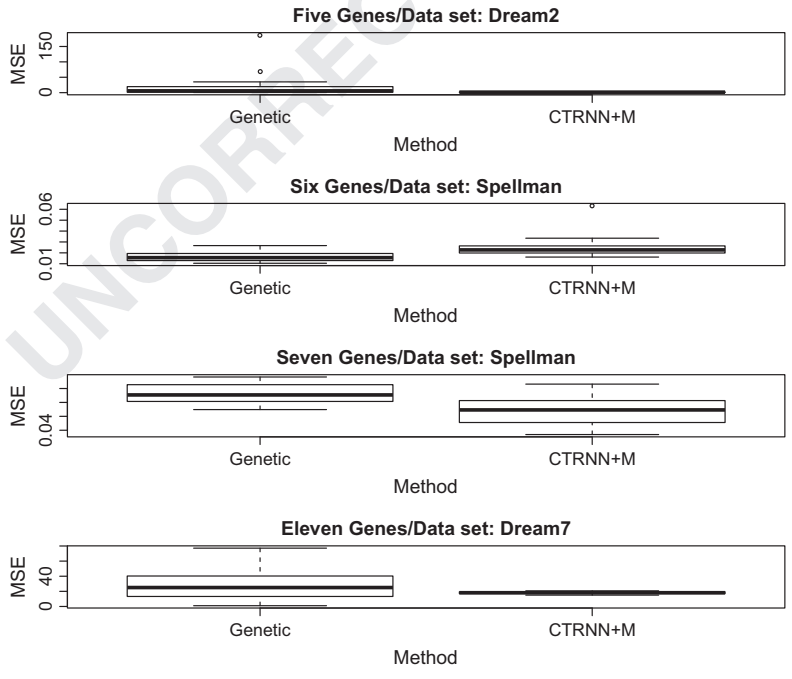


Fig. 6. Comparison of MSE for different methods for: 5 gene data set from DREAM2; 6, 7 from Spellman data set and 11 from DREAM7. ANOVA tests for each one gives a $p < 0.0001$ for each one, which means that the means of the methods are different between them.

not necessary to integrate the network dynamics for the purpose of parameter optimization. However a full Bayesian treatment, in which the parameter's estimations are drawn, for instance, from the sampling of the posterior, is intended as a future work.

Acknowledgement

This work was partially supported by the National Council of Science and Technology of Mexico under grant CONACYT CB-167651 and UANL-PAICYT.

Please cite this article as: A. Berrones, et al., Parameter inference of general nonlinear dynamical models of gene regulatory networks from small and noisy time series, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.10.095

Appendices

Tables 1 and 2 .

Table 1
Analysis of variance for MSE for factorial experiment.

Source	GL	SS	SS Adjust.	Mean SS Adj.	*F*	*P*
SampleRep	2	2617	2617	1309	0.48	0.620
SampleTemp	2	6936	6936	3468	1.27	0.282
Noise	2	9867	9867	4933	1.8	0.165
StepSize	3	13,373	13,373	4458	1.63	0.181
Method	1	859,042	859,042	859,042	313.93	0.000
SampleRep∗SampleTemp	4	21,460	21,460	5365	1.96	0.099
SampleRep∗Noise	4	1677	1677	419	0.15	0.962
SampleRep∗StepSize	6	18,892	18,892	3149	1.15	0.331
SampleRep∗Method	2	3797	3797	1899	0.69	0.500
SampleTemp∗Noise	4	7770	7770	1942	0.71	0.585
SampleTemp∗StepSize	6	13,651	13,651	2275	0.83	0.546
SampleTemp∗Method	2	10,593	10,593	5297	1.94	0.145
Noise∗StepSize	6	10,605	10,605	1768	0.65	0.693
Noise∗Method	2	11,459	11,459	5730	2.09	0.124
StepSize∗Method	3	14,210	14,210	4737	1.73	0.159
SampleRep∗SampleTemp∗Noise	8	16,109	16,109	2014	0.74	0.660
SampleRep∗SampleTemp∗StepSize	12	31,705	31,705	2642	0.97	0.480
SampleRep∗SampleTemp∗Method	4	20,411	20,411	5103	1.86	0.115
SampleRep∗Noise∗StepSize	12	27,974	27,974	2331	0.85	0.597
SampleRep∗Noise∗Method	4	4392	4392	1098	0.4	0.808
SampleRep∗StepSize∗Method	6	14,064	14,064	2344	0.86	0.526
SampleTemp∗Noise∗StepSize	12	37,432	37,432	3119	1.14	0.324
SampleTemp∗Noise∗Method	4	12,182	12,182	3045	1.11	0.349
SampleTemp∗StepSize∗Method	6	12,329	12,329	2055	0.75	0.609
Noise∗StepSize∗Method	6	8460	8460	1410	0.52	0.797
SampleRep∗SampleTemp∗Noise∗StepSize	24	65,954	65,954	2748	1	0.457
SampleRep∗SampleTemp∗Noise∗Method	8	14,431	14,431	1804	0.66	0.728
SampleRep∗SampleTemp∗StepSize∗Method	12	33,486	33,486	2791	1.02	0.428
SampleRep∗Noise∗StepSize∗Method	12	21,144	21,144	1762	0.64	0.805
SampleTemp∗Noise∗StepSize∗Method	12	36,883	36,883	3074	1.12	0.337
SampleRep∗SampleTemp∗Noise∗StepSize∗Method	24	69,601	69,601	2900	1.06	0.385
Error	864	2,364,251	2,364,251	2736		
Total	1079	3,796,758				

Table 2
Analysis of variance for time for factorial experiment.

Source	GL	SS	SS Adjust.	Mean SS Adj.	*F*	*P*
SampleRep	2	7730	7730	3865	4.23	0.015
SampleTemp	2	67,458	67,458	33,729	36.95	0.000
Noise	2	59,349	59,349	29,675	32.51	0.000
StepSize	3	181,382,308	181,382,308	60,460,769	66228.9	0.000
Method	1	1,323,363	1,323,363	1,323,363	1449.62	0.000
SampleRep∗SampleTemp	4	31,325	31,325	7831	8.58	0.000
SampleRep∗Noise	4	24,745	24,745	6186	6.78	0.000
SampleRep∗StepSize	6	60,987	60,987	10,164	11.13	0.000
SampleRep∗Method	2	1598	1598	799	0.88	0.417
SampleTemp∗Noise	4	118,056	118,056	29,514	32.33	0.000
SampleTemp∗StepSize	6	19,482	19,482	3247	3.56	0.002
SampleTemp∗Method	2	171	171	86	0.09	0.910
Noise∗StepSize	6	483,680	483,680	80,613	88.3	0.000
Noise∗Method	2	10920	10,920	5460	5.98	0.003
StepSize∗Method	3	4,868,088	4,868,088	1,622,696	1777.51	0.000
SampleRep∗SampleTemp∗Noise	8	84,978	84,978	10,622	11.64	0.000
SampleRep∗SampleTemp∗StepSize	12	86,440	86,440	7203	7.89	0.000
SampleRep∗SampleTemp∗Method	4	514	514	128	0.14	0.967
SampleRep∗Noise∗StepSize	12	139,301	139,301	11,608	12.72	0.000
SampleRep∗Noise∗Method	4	1708	1708	427	0.47	0.759
SampleRep∗StepSize∗Method	6	4239	4239	707	0.77	0.590
SampleTemp∗Noise∗StepSize	12	477,384	477,384	39,782	43.58	0.000
SampleTemp∗Noise∗Method	4	14,514	14,514	3628	3.97	0.003
SampleTemp∗StepSize∗Method	6	1593	1593	265	0.29	0.941
Noise∗StepSize∗Method	6	57,004	57,004	9501	10.41	0.000
SampleRep∗SampleTemp∗Noise∗StepSize	24	161,404	161,404	6725	7.37	0.000
SampleRep∗SampleTemp∗Noise∗Method	8	5840	5840	730	0.8	0.603
SampleRep∗SampleTemp∗StepSize∗Method	12	3886	3886	324	0.35	0.978
SampleRep∗Noise∗StepSize∗Method	12	4138	4138	345	0.38	0.971
SampleTemp∗Noise∗StepSize∗Method	12	35,370	35,370	2948	3.23	0.000
SampleRep∗SampleTemp∗Noise∗StepSize∗Method	24	16,820	16,820	701	0.77	0.780
Error	864	788,751	788,751	913		
Total	1079	190,343,145				

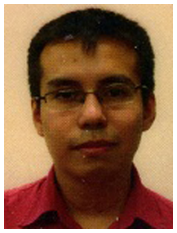
References

[1] Christopher M. Bishop, Pattern Recognition and Machine Learning, vol. 1, Springer, New York, 2006.
[2] Radford M. Neal, Bayesian Learning for Neural Networks, University of Toronto, Diss, 1995.
[3] David JC. MacKay, Information Theory, Inference, and Learning Algorithms, vol. 7, Cambridge University Press, Cambridge, 2003.
[4] J. Mazur, D. Ritter, G. Reinelt, L. Kaderali, Reconstructing nonlinear dynamic models of gene regulation using stochastic sampling, BMC Bioinform. 10 (1) (2009) 448.
[5] Y.X. Wang, H. Huang, Review on statistical methods for gene network reconstruction using expression data, J. Theor. Biol. (2014).
[6] D. Marbach, R.J. Prill, T. Schaffter, C. Mattiussi, D. Floreano, G. Stolovitzky, Revealing strengths and weaknesses of methods for gene network inference, Proc. Natl. Acad. Sci. 107 (14) (2010) 6286–6291.
[7] D. Marbach, J.C. Costello, R. Küffner, N.M. Vega, R.J. Prill, D.M. Camacho, Wisdom of crowds for robust gene network inference, Nat. Methods 9 (8) (2012) 796–804 , DREAM5 Consortium.
[8] R. Xu, G.K. Venayagamoorthy, D.C. Wunsch II, Modeling of gene regulatory networks with hybrid differential evolution and particle swarm optimization, Neural Netw. 20 (8) (2007) 917–927.
[9] J. Sun, J.M. Garibaldi, C. Hodgman, Parameter estimation using metaheuristics in systems biology: a comprehensive review, IEEE/ACM Trans. Comput. Biol. Bioinform. 9 (1) (2012) 185–202.
[10] J.M. Varah, A spline least squares method for numerical parameter estimation in differential equations, SIAM J. Sci. Stat. Comput. 3 (1) (1982) 28–46.
[11] P. D'haeseleer, S. Liang, R. Somogyi, Genetic network inference: from co-expression clustering to reverse engineering, Bioinformatics 16 (8) (2000) 707–726.
[12] K. Raza, M. Alam, R. Parveen, 2014. Recurrent Neural Network Based Hybrid Model of Gene Regulatory Network. arXiv:1408.5405.
[13] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, N. Friedman, Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data, Nat. Genet. 34 (2) (2003) 166–176.
[14] F.M. Lopes, E.A. de Oliveira, R.M. Cesar, Inference of gene regulatory networks from time series by Tsallis entropy, BMC Syst. Biol. 5 (1) (2011) 61.
[15] K.H. Cho, S.M. Choo, S.H. Jung, J.R. Kim, H.S. Choi, J. Kim, Reverse engineering of gene regulatory networks, IET Syst. Biol. 1 (3) (2007) 149–163.
[16] J. Vohradsky, Neural model of the genetic network, J. Biol. Chem. 276 (39) (2001) 36168–36173.
[17] R.M. Neal, Bayesian Learning for Neural Network, vol. 118, Springer, New York, 1996.
[18] M.B. Elowitz, S. Leibler, A synthetic oscillatory network of transcriptional regulators, Nature 403 (6767) (2000) 335–338.
[19] O. Buse, R. Pérez, A. Kuznetsov, Dynamical properties of the repressilator model, Phys. Rev. E 81 (6) (2010) 066206.
[20] S.A. Thomas, Y. Jin, Reconstructing biological gene regulatory networks: where optimization meets big data, Evol. Intell. 7 (1) (2014) 29–47.
[21] J.K. Lee, Statistical Bioinformatics: For Biomedical and Life Science Researchers, John Wiley & Sons, 2011.
[22] Luca Scrucca, GA: a package for genetic algorithms in R, J. Stat. Softw. 53 (4) (2013) 1–37 , URL ⟨http://www.jstatsoft.org/v53/i04/.
[23] A. Sîrbu, H.J. Ruskin, M. Crane, Comparison of evolutionary algorithms in gene regulatory network model inference, BMC Bioinform. 11 (1) (2010) 59, http://dx.doi.org/10.1186/1471-2105-11-59, Data set.
[24] G.A. Ruz, E. Goles, Learning gene regulatory networks using the bees algorithm, Neural Comput. Appl. 22 (1) (2013) 63–70.
[25] G. Stolovitzky, R.J. Prill, A. Califano, Lessons from the DREAM2 challenges, Ann. N. Y. Acad. Sci. 1158 (2009) 159–195.
[26] I. Cantone, L. Marucci, F. Iorio, M.A. Ricci, V. Belcastro, M. Bansal, M.P. Cosma, A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches, Cell 137 (1) (2009) 172–181.
[27] M. Kabir, N. Noman, H. Iba, Reverse engineering gene regulatory network from microarray data using linear time-variant model, BMC Bioinform. 11 (Suppl. 1) (2010) S56.
[28] D. Tominaga, N. Koga, M. Okamoto, Efficient numerical optimization algorithm based on genetic algorithm for inverse problem, in: Proceedings of Genetic and Evolutionary Computation Conference, pp. 251–258.
[29] S. Kimura, K. Ide, A. Kashihara, M. Kano, M. Hatakeyama, R. Masui, A. Konagaya, Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm, Bioinformatics 21 (7) (2005) 1154–1163.
[30] S. Kimura, K. Sonoda, S. Yamane, H. Maeda, K. Matsumura, M. Hatakeyama, Function approximation approach to the inference of reduced NGnet models of genetic networks, BMC Bioinform. 9 (1) (2008) 23.
[31] N. Noman, H. Iba, Inferring gene regulatory networks using differential evolution with local search heuristics, IEEE/ACM Trans. Comput. Biol. Bioinform. (TCBB) 4 (4) (2007) 634–647.
[32] S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi, M. Tomita, Dynamic modeling of genetic networks using genetic algorithm and S-system, Bioinformatics 19 (5) (2003) 643–650.
[33] B. Bauer, M. Reynolds, Recovering data from scanned graphs: performance of Frantz's g3data software, Behav. Res. Methods 40 (3) (2008) 858–868.
[34] P. Meyer, T. Cokelaer, D. Chandran, K.H. Kim, P.R. Loh, G. Tucker, J. Saez-Rodriguez, Network topology and parameter estimation: from experimental design methods to gene regulatory network kinetics using a community based approach, BMC Syst. Biol. 8 (1) (2014) 13.

Arturo Berrones received a Ph.D. degree in Physics in 2002 from Universidad Autonoma del Edo, de Morelos, Mexico. In 2003 he was a postdoctoral research assistant at the University of Florence, Italy, working in the field of complex systems. Since 2004, he has been an assistant professor of the Systems Engineering Program, at Universidad Autonoma de Nuevo Leon, Mexico. His current research interests are the statistical and computational aspects of complex systems and the interface between statistical mechanics and artificial intelligence.

María Aracelia Alcorta García received a Ph.D. degree in Physical Industrial Engineering graduate program in 2003 from Universidad Autonoma de Nuevo león, México. In 2005–2006 she did a postdoctoral stay in the University of California in San Diego. Since 2004, she has been a titular professor of the Physical Industrial Engineering Program, at Universidad Autonoma de Nuevo Leon, Mexico. She is a founder of Graduate Program in sciences with Orientation in Mathematics, in 2010, in Universidad Autonoma de Nuevo Leon, Facultad de Ciencias Físico Matemáticas. She was coordinator of this from 2010 to 2013. She has research interests in Mathematics applications, specificaly of non-linear risk-sensitive stochastic control, and Mechatronics.

Javier Almaguer received a Ph.D. in Physics at the Autonomous University of Morelos, in Cuernavaca. He is currently a full-time research professor in Universidad Autonoma de Nuevo Leon, Facultad de Ciencias Físico Matemáticas. His current research interests are Mathematical modeling and applications using complex systems, dynamic equilibrium of complex systems, heuristic search and collective phenomena.



Edgar Jimenez received a M.S in Applied Statistics at Instituto Tecnológico y de Estudios Superiores de Monterrey in 2004. Currently in doing a Ph.D. in Sciences with Orientation in Mathematics at Universidad Autonoma de Nuevo león, México. His research interests are complex systems, dynamical process in networks and optimization.

Please cite this article as: A. Berrones, et al., Parameter inference of general nonlinear dynamical models of gene regulatory networks from small and noisy time series, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.10.095

Apéndice 2

Tablas de análisis de varianza para el experimento

Table 5.1: Análisis de varianza para el ECM para el experimento factorial

Fuente	GL	SC	SC Ajust.	Media SC Aj.	F	P
Repeticiones Experimento	2	2617	2617	1309	0.48	0.620
MuestrasTemporales	2	6936	6936	3468	1.27	0.282
Ruido	2	9867	9867	4933	1.8	0.165
Tamaño Paso	3	13373	13373	4458	1.63	0.181
Método	1	859042	859042	859042	313.93	0.000
Repeticiones Experimento * MuestrasTemporales	4	21460	21460	5365	1.96	0.099
Repeticiones Experimento * Ruido	4	1677	1677	419	0.15	0.962
Repeticiones Experimento * Tamaño Paso	6	18892	18892	3149	1.15	0.331
Repeticiones Experimento * Método	2	3797	3797	1899	0.69	0.500
Muestras Temporales * Ruido	4	7770	7770	1942	0.71	0.585
Muestras Temporales * Tamaño Paso	6	13651	13651	2275	0.83	0.546

Table 5.1: Análisis de varianza para el ECM para el experimento factorial

Fuente	GL	SC	SC Ajust.	Media SC Aj.	F	P
Muestras Temporales * Método	2	10593	10593	5297	1.94	0.145
Ruido * Tamaño Paso	6	10605	10605	1768	0.65	0.693
Ruido * Método	2	11459	11459	5730	2.09	0.124
Tamaño Paso * Método	3	14210	14210	4737	1.73	0.159
Repeticiones Experimento * Muestras Temporales * Ruido	8	16109	16109	2014	0.74	0.660
Repeticiones Experimento * Muestras Temporales * Tamaño Paso	12	31705	31705	2642	0.97	0.480
Repeticiones Experimento * Muestras Temporales * Método	4	20411	20411	5103	1.86	0.115
Repeticiones Experimento * Ruido * Tamaño Paso	12	27974	27974	2331	0.85	0.597
Repeticiones Experimento * Ruido * Método	4	4392	4392	1098	0.4	0.808
Repeticiones Experimento * Tamaño Paso * Método	6	14064	14064	2344	0.86	0.526

Table 5.1: Análisis de varianza para el ECM para el experimento factorial

Fuente	GL	SC	SC Ajust.	Media SC Aj.	F	P
Muestras Temporales * Ruido * Tamaño Paso	12	37432	37432	3119	1.14	0.324
Muestras Temporales * Ruido * Método	4	12182	12182	3045	1.11	0.349
MuestrasTemporales * TamañoPaso * Método	6	12329	12329	2055	0.75	0.609
Ruido * TamañoPaso * Método	6	8460	8460	1410	0.52	0.797
Repeticiones Experimento * Muestras Temporales * Ruido * Tamaño Paso	24	65954	65954	2748	1	0.457
Repeticiones Experimento * Muestras Temporales * Ruido * Método	8	14431	14431	1804	0.66	0.728
Repeticiones Experimento * Muestras Temporales * Tamaño Paso * Método	12	33486	33486	2791	1.02	0.428

Table 5.1: Análisis de varianza para el ECM para el experimento factorial

Fuente	GL	SC	SC Ajust.	Media SC Aj.	F	P
Repeticiones Experimento * Ruido * Tamaño Paso * Método	12	21144	21144	1762	0.64	0.805
Muestras Temporales * Ruido * Tamaño Paso * Método	12	36883	36883	3074	1.12	0.337
Repeticiones Experimento * Muestras Temporales * Ruido * Tamaño Paso * Método	24	69601	69601	2900	1.06	0.385
Error	864	2364251	2364251	2736		
Total	1079	3796758				

Table 5.2: Análisis de varianza para el el tiempo en el experimento factorial

Fuente	GL	SC	SC Ad-just.	Media SC Aj.	F	P
Repeticiones Experimento	2	7730	7730	3865	4.23	0.015
Muestras Temporales	2	67458	67458	33729	36.95	0.000
Ruido	2	59349	59349	29675	32.51	0.000
Tamaño Paso	3	181382308	181382308	60460769	66228.9	0.000
Método	1	1323363	1323363	1323363	1449.62	0.000
Repeticiones Experimento * Muestras Temporales	4	31325	31325	7831	8.58	0.000
Repeticiones Experimento * Ruido	4	24745	24745	6186	6.78	0.000
Repeticiones Experimento * Tamaño Paso	6	60987	60987	10164	11.13	0.000
Repeticiones Experimento * Método	2	1598	1598	799	0.88	0.417
Muestras Temporales * Ruido	4	118056	118056	29514	32.33	0.000
Muestras Temporales * Tamaño Paso	6	19482	19482	3247	3.56	0.002

Table 5.2: Análisis de varianza para el el tiempo en el experimento factorial

Fuente	GL	SC	SC Ad-just.	Media SC Aj.	F	P
Muestras Temporales * Método	2	171	171	86	0.09	0.910
Ruido * Tamaño Paso	6	483680	483680	80613	88.3	0.000
Ruido * Método	2	10920	10920	5460	5.98	0.003
Tamaño Paso * Método	3	4868088	4868088	1622696	1777.51	0.000
Repeticiones Experimento * Muestras Temporales * Ruido	8	84978	84978	10622	11.64	0.000
Repeticiones Experimento* Muestras Temporales * Tamaño Paso	12	86440	86440	7203	7.89	0.000
Repeticiones Experimento * Muestras Temporales * Método	4	514	514	128	0.14	0.967
Repeticiones Experimento * Ruido * Tamaño Paso	12	139301	139301	11608	12.72	0.000
Repeticiones Experimento * Ruido * Método	4	1708	1708	427	0.47	0.759

Table 5.2: Análisis de varianza para el el tiempo en el experimento factorial

Fuente	GL	SC	SC Ad-just.	Media SC Aj.	F	P
Repeticiones Experimento * Tamaño Paso * Método	6	4239	4239	707	0.77	0.590
Muestras Temporales * Ruido * Tamaño Paso	12	477384	477384	39782	43.58	0.000
Muestras Temporales * Ruido * Método	4	14514	14514	3628	3.97	0.003
Muestras Temporales * Tamaño Paso * Método	6	1593	1593	265	0.29	0.941
Ruido * Tamaño Paso * Método	6	57004	57004	9501	10.41	0.000
Repeticiones Experimento * Muestras Temporales * Ruido * Tamaño Paso	24	161404	161404	6725	7.37	0.000
Repeticiones Experimento * Muestras Temporales * Ruido * Método	8	5840	5840	730	0.8	0.603

Table 5.2: Análisis de varianza para el el tiempo en el experimento factorial

Fuente	GL	SC	SC Ad-just.	Media SC Aj.	F	P
Repeticiones Experimento * Muestras Temporales * Tamaño Paso * Método	12	3886	3886	324	0.35	0.978
Repeticiones Experimento * Ruido * Tamaño Paso * Método	12	4138	4138	345	0.38	0.971
Muestras Temporales * Ruido * Tamaño Paso * Método	12	35370	35370	2948	3.23	0.000
Repeticiones Experimento * Muestras Temporales * Ruido * Tamaño Paso*Método	24	16820	16820	701	0.77	0.780
Error	864	788751	788751	913		
Total	1079	190343145				

Apéndice 3

Abreviaturas usadas en el documento

Abreviatura	Significado
GRN	Red de regulación genética
mRNA	ARN mensa
CLAIO	Congreso Latinoamericano de Investigación de Operaciones
PSO	Optimización basada en enjambre de partículas
ChIP	Inmonuprecipitación de cromatina
BPTT	Propagación hacia atrás del tiempo
CTRNN	Red neuronal recurrente en tiempo continuo
MAP	Estimación máxima a posteriori
LOESS	Regresión local
ANOVA	Análisis de Varianza